

ISSN 0868-6157

КОМПЬЮТЕР ПРЕСС

LAND

AZTECH

9'92

Трудно отказаться...



Совместное советско-американское предприятие «СОВАМИНКО»

КОМПЬЮТЕР ПРЕСС

9'92



КОМПЬЮТЕРПРЕСС

Издается с 1989 года
Выходит 12 раз в год
9'92 (33)

Адрес редакции:
113093 Москва, аб.ящик 37
Факс: (095) 200-22-89
Телефон для справок: (095) 471-32-63
E-mail: editorial@computerpress.msk.su

Сдано в набор 13.07.92. Подписано к печати 24.07.92.
Формат 84x108/16. Печать офсетная. Бумага типографская. Усл.печ.листов 8,4+0,42 (обл.).
Тираж 62000 экз. Заказ 3010. С-9.

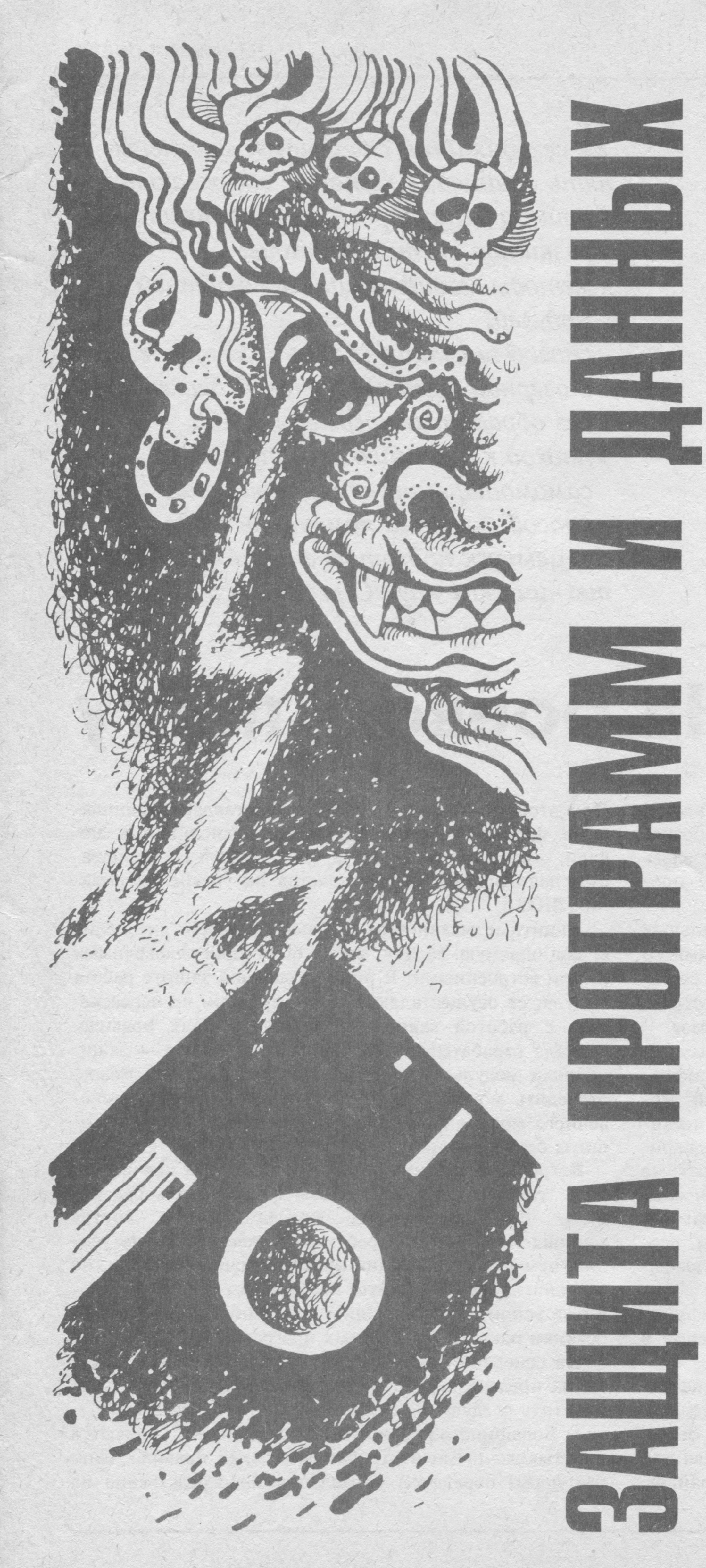
Оригинал-макет подготовлен агентством «КомпьютерПресс».

Тексты проверены системой «ОРФО».

Журнал сверстан на оборудовании фирмы Summit Systems.

Отпечатано в полиграфической фирме «Красный пролетарий» РГИИЦ «Республика».
103473 Москва, И-473, Краснопролетарская, 16.

СПЕЦИАЛЬНЫЙ ВЫПУСК



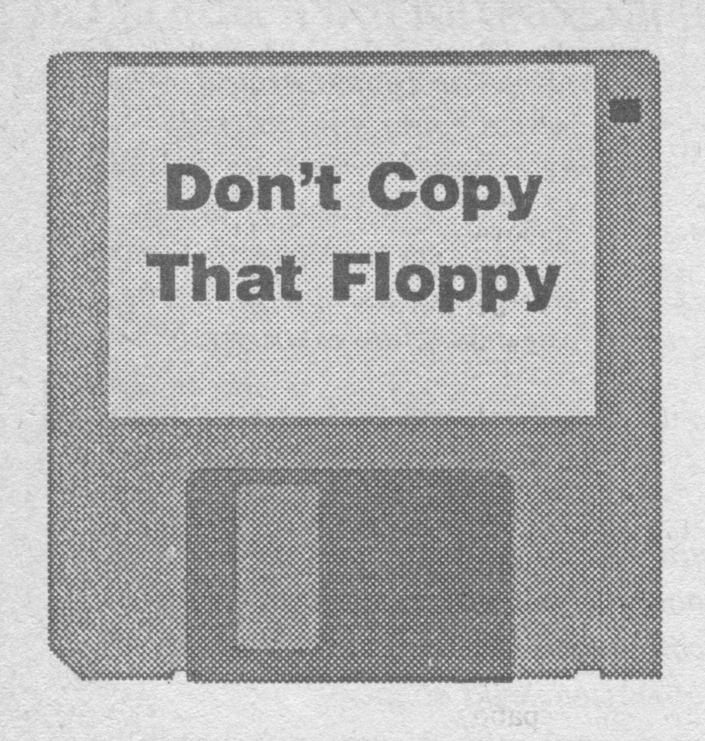
ЗАЩИТА ПРОГРАММ И ДАННЫХ

"Кто не горит душой — тому не раскрою ничего; кто не скорбит, того не разовью. И ничего не отвечу тому, кто не скажет ни слова о трех сторонах квадрата, когда ему объяснен один угол".

Из трактата древнекитайского философа

Защита программ и данных программно-аппаратными способами представляет собой сегодня активно разрабатываемую область программирования. У истоков этого начинания стоят энтузиасты, которым красивое решение необычных задач доставляет истинное наслаждение. Большинство авторов сегодняшнего спецвыпуска занимаются разработкой защитных механизмов не только по профессиональной необходимости. Возможно, объясняется это тем, что, как утверждал Бернард Шоу: "Мозг человека удивительный орган: он начинает работать как только вы просыпаетесь утром, и не прекращает до тех пор, пока вы не явитесь к себе на службу". Сделанные наработки со временем становятся основой нарождающейся теории, алгоритмы классифицируются по различным критериям и возникает необходимость обмена информацией для выбора направлений дальнейших исследований. Кроме того, поделиться достигнутым — это естественная потребность каждого нормального человека.

Своими публикациями мы не стремимся "научить рыбу плавать", но очень надеемся, что заинтересованный читатель найдет здесь для себя конкретные приложения. Первое знакомство начнем с обзорной экскурсии, после которой перейдем к конкретным задачам, приемам и алгоритмам.



В сегодняшний спецвыпуск включены пять статей, в которых обсуждаются:

- *принципы построения и перспективы развития систем защиты;*
- *методы модификации исполняемых модулей;*
- *способ эмуляции сбойных секторов (лазерной метки) ключевой дискеты без обработки прерывания 13h;*
- *подход к созданию бесконечно самомодифицирующейся задачи;*
- *способ скрытно пометить файл.*

Спецвыпуск подготовлен кандидатом технических наук Сергеем Расторгуевым.

На боевом посту

Существуют различные подходы к защите от несанкционированного копирования информации, обрабатываемой на ЭВМ. Мы будем рассматривать здесь только программно-аппаратные способы защиты программ и данных.

На основании анализа более чем 10 специальных систем, ориентированных на защиту программного обеспечения от НСК [1], можно сказать, что на сегодня общепринятым стал следующий подход к построению программно-аппаратных защитных механизмов.

Данные шифруются или архивируются любым из возможных алгоритмов, а сам алгоритм шифрации/дешифрации встраивается в исполняемый модуль, который в свою очередь защищается от несанкционированного копирования посредством предварительной настройки или на программно-аппаратные особенности ПЭВМ, и/или на психофизиологические особенности оператора, работающего с программой (различные парольные системы, тестовые ответы, особенности работы за клавиатурой, особенности владения (подписи) мышкой).

Все защитные механизмы могут быть классифицированы по способу отработки и способу внедрения в защищаемый модуль.

Защитные механизмы могут отрабатывать в пакетном режиме и/или иметь резидентную часть, то есть постоянно находиться в оперативной памяти, отслеживая все операции чтения/записи и осуществляя при этом шифрацию/дешифрацию защищаемых файлов. При этом неважно, что собой представляют защищаемые файлы — шрифты, данные, исполняемые модули, изображения и т.д. — поскольку шифрация/дешифрация осуществляется на уровне прерываний BIOS.

Защитные механизмы по способу своего внедрения в защищаемую задачу могут быть пристыковочными и/или встроенными. В пристыковочной защите работа модуля, ее осуществляющего, по времени не пересекается с работой защищаемого модуля. Как правило, сначала отрабатывает модуль защиты, затем — защищенный модуль. В этом случае, как правило, можно отследить момент передачи управления от пристыковочного модуля к защищенному и обойти систему защиты без ее анализа.

Встроенная защита может отрабатывать как до начала, так и в процессе работы функционального модуля. Для встраивания модуля защиты внутрь защищаемого модуля требуются более серьезные усилия, чем для встраивания пристыковочного модуля, но для снятия такой защиты необходимы не менее серьезные усилия. Такую защиту часто пишут сами разработчики пакетов прикладных программ.

На основании анализа существующих систем защиты на предмет вскрытия защитных механизмов можно отметить следующие существенные их недостатки.

1. Большинство защитных механизмов являются пристыковочными и отрабатывают, как правило, один раз перед передачей управления непосредственно на

исполняемый код защищенной задачи. Это значит, что существует принципиальная возможность исключить из задачи пристыковочную часть. Сделать это можно следующим образом:

1) разрабатывается резидентная задача, которая либо при нажатии соответствующих "горячих" клавиш, либо при получении заданного прерывания и/или функции прерывания копирует содержимое оперативной памяти в файл на диске;
2) перед запуском защищенной задачи в память загружается разработанная резидентная программа;
3) после отработки защитного механизма, когда управление уже передано непосредственно задаче, активизируется резидентная программа и копирует исполняемый код защищенной задачи вместе с текущим значением всех регистров на диск.

Запуск созданного таким образом файла осуществляется в обратном порядке. Специальная задача загружает файл в оперативную память, устанавливает зафиксированные в файле значения регистров и передает управление задаче.

Насколько нам известно, подобные программы уже существуют и активно рекламируются (утилиты CATCHER — центром САПР Ассоциации "Наука" и EXEB — Центром СОВМАРКЕТ), и с этим ничего поделать нельзя — время требует иного подхода, ибо, как утверждал классик, "каждому времени своя женщина".

2. Отсутствует возможность теоретически оценить надежность системы защиты. Сегодня качество защитных механизмов того или иного пакета оценивается субъективно — по времени, требуемому для снятия защиты с защищенной программы конкретному человеку.

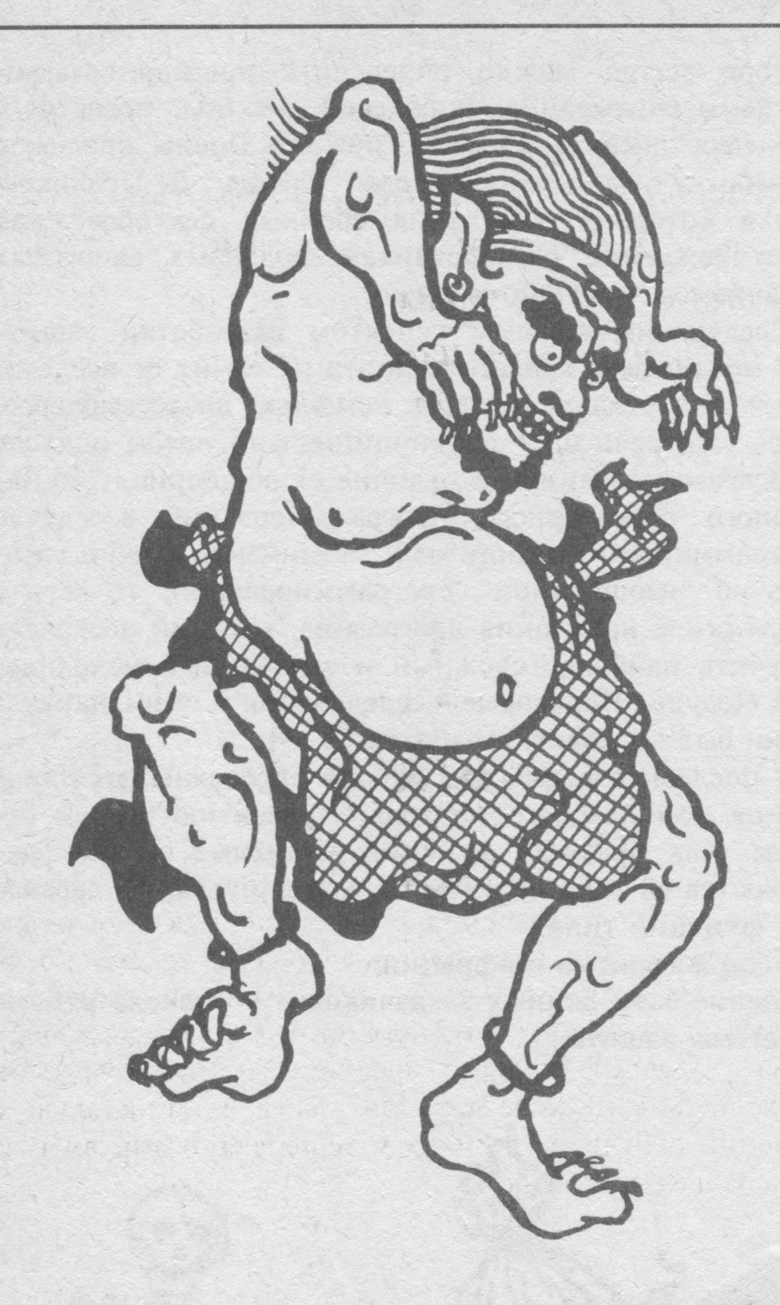
3. В архитектуре большинства типов ПЭВМ не существует такой аппаратной особенности, на основании анализа которой в программе можно было бы сделать вывод о факте НСК, а это особенно важно в случае поставки программного обеспечения (ПО) заказчику по почте. Использование для последующей привязки ПО к ПЭВМ аппаратных заглушек или специальным образом физически помеченных дискет неэффективно, так как сейчас уже хорошо проработаны способы вскрытия такой защиты: для аппаратной заглушки используют принцип спаривания компьютеров, подключая через "прозрачную" аппаратную заглушку второй компьютер, который отслеживает все передаваемые между заглушкой и ПЭВМ сигналы; для физически испорченной дискеты — путем захвата специальной резидентной программой соответствующего аппаратного прерывания от устройства и подмены выдаваемых им кодов возврата на требуемые; дефектные сектора всегда можно определить предварительным тестовым считыванием ключевой дискеты, после чего ключевая дискета уже не нужна. Очень красивую разработку на данную тему сделал Д. Правиков, статья которого "Эмуляция сбойных секторов" как предупреждение разработчикам подобных защитных механизмов, приводится ниже.

Следующим важным аспектом разработки защитных механизмов является защита их самих от исследований под отладчиком и с помощью дизассемблеров. Здесь уже речь идет о принципиально новом подходе к программированию. В отличие от общепринятого наглядного структурного программирования в случае программирования защитных механизмов нужно говорить об "изощренном" программировании, то есть о таком стиле написания программы, который позволяет получить наиболее сложный и запутанный исполняемый модуль. Некоторые исследования в этом направлении были сделаны в работах [2, 3].

В последнее время все больше прослеживается тенденция комплексного подхода к развитию систем защиты. Как правило, функции защитных систем расширяются за счет включения дополнительных сервисных функций типа:

- выбор алгоритма шифрации;
- ведение базы данных заказчиков и протокола работы системы защиты;





- возможность выбора алгоритма защиты от НСК;
- защита от НСД ПЭВМ;
- восстановление удаленных данных;
- парольные системы идентификации пользователя;
- возможность удаления без восстановления,

и других.

Что касается основных функций, то тенденция развития программно-аппаратных систем защиты от НСК связана с преодолением названных выше недостатков. В связи с этим основные работы по развитию программно-аппаратных систем защиты, на наш взгляд, будут вестись по следующим направлениям.

1. Организация профессиональной защиты ПО от исследований (подробнее см. [2]).

2. Выработка механизма для объективной оценки надежности системы защиты (подробнее см. [4]).

3. Создание встроенных защитных механизмов.

Основная трудность при разработке алгоритмов встраивания пристыковочных защитных механизмов — определение места (нескольких мест) непосредственно в теле исполняемого модуля, где можно произвести замены кодов команд. Для решения этой задачи в составе системы защиты желательно иметь свой дизассемблер и соответствующий логический блок, анализирующий возможную частоту повторов соответствующей части защищаемой задачи. Иначе возможна ситуация, когда защита будет встроена в ту часть защищаемой задачи, на которую очень редко передается управление (или вообще не передается), например, в блок обработки ошибок. Названная проблема относится к теории алгоритмов и практическое ее решение на сегодняшний день еще очень далеко от совершенства.

Проблема включения собственного модуля в защищаемый пакет может рассматриваться не только в плане решения конкретной задачи внедрения защитного механизма, но также и с точки зрения возможной КОРРЕКЦИИ ИСПОЛНЯЕМЫХ МОДУЛЕЙ (подробности в соответствующей статье).

4. Идентификация пользователя. Получение устойчивых характеристик почерка оператора, работающего с клавиатурой и/или с мышкой.

5. Ориентация на аппаратные особенности устройств, в частности, на закон распределения ошибок и т.п.

Как показала практика работы непосредственно с устройствами, динамические характеристики различных частей компьютера и их взаимные соотношения не являются достаточно устойчивыми, чтобы на них опираться. Для повышения устойчивости рекомендуется использовать закон распределения динамических характеристик, то есть постоянно их отслеживать и в случае допустимого порога изменения корректировать первоначальные математические ожидания и дисперсии.

Для создания ключевой дискеты подобный подход предполагает организацию записи на дискету с пониженным уровнем записи логической единицы на специальном дисководе с последующим многократным чтением в нормальных условиях на предмет определения закона распределения ошибок для сбойных секторов. После этого закон распределения ошибок (мат. ожидание и дисперсия для каждого сектора) фиксируется в защищаемой программе при настройке ее на ключевую дискету. Даже при аналоговом копировании подобной дискеты законы распределения ошибок на исходной и сдублированной дискете, на наш взгляд, должны различаться.

С.Расторгуев

Использованы материалы:

1. Расторгуев С.П., Дмитриевский Н.Н. Искусство защиты и “раздевания“ программ. — М.: Центр СОВМАРКЕТ, 1991.
2. Долгин А.Е., Расторгуев С.П. Защита программ от дизассемблеров и отладчиков. КомпьютерПресс, 1992, №4.
3. Долгин А.Е. Защита программного обеспечения от исследования. Электронный журнал “НСК“, 1992, №1. — М.: Центр СОВМАРКЕТ.
4. Расторгуев С.П. Купить или украсть? Оценка защиты. КомпьютерПресс, 1992, №8.



Коррекция работы исполняемых модулей при отсутствии исходных текстов программ — "высший пилотаж" в программировании. Методы внесения изменений в готовые EXE-файлы обсуждаются в этой статье.

Коррекция исполняемых модулей

"Бывает только неправильный путь, но не бывает безвыходного положения."

Восточная мудрость

К одной из наиболее сложных задач в программировании относится доработка или модификация функций исполняемых задач, для которых отсутствуют исходные тексты, а сами задачи защищены от исследования [1]. Это достаточно узкая область программирования, и, возможно, немногим из наших читателей приходилось заниматься подобными работами. Мы сами столкнулись с данной проблемой, когда возникла необходимость исправить выявленную пользователем ошибку в нашем пакете, исходные тексты которого, хранимые нами на винчестере, были нечаянно уничтожены во время борьбы с вирусом. Переписывать весь пакет за тот срок, в течение которого мы гарантировали своим пользователям устранение любых замечаний, представлялось невозможным. Поэтому пришлось искать другие пути модификации задачи. В результате был создан определенный подход и даже разработан набор инструментальных средств, который мы с успехом применяем с тех пор.

Данная статья посвящена методам корректировки исполняемых модулей при отсутствии исходных текстов программ. Статья ориентирована в первую очередь на разработчиков программного обеспечения для ПЭВМ типа IBM PC в среде DOS.

Операционная система DOS поддерживает два типа исполняемых модулей: COM- и EXE-файлы. Мы рассмотрим только EXE-файлы, как наиболее популярные и более сложные для модификации. В то же время большая часть сказанного может быть отнесена и к COM-файлам.

Для удобства восприятия материала и приводимых примеров кратко опишем алгоритм запуска и основные особенности выполнения задач в среде DOS.

EXE-файл условно может быть разделен на заголовок, таблицу перемещения и собственно исполняемый код. Структура заголовка приведена в тексте представленного примера в конце статьи. Информация заголовка используется DOS для загрузки модуля в оперативную память, корректировки сегментных адресов согласно таблице перемещения, присвоения начальных значений основным регистрам и передачи управления непосредственно задаче.

В процессе выполнения задачи все основные функции, связанные с вводом/выводом, захватом/освобождением оперативной памяти и т.п., отрабатываются через 21h прерывание DOS, а значит всегда могут быть доработаны и модифицированы с помощью специальной задачи, которая либо резидентна в памяти, либо непосредственно включена в модифицируемый EXE-файл. В статье мы рассмотрим второй подход, как более удобный для конечного пользователя.

Для того чтобы иметь возможность откорректировать функции EXE-файла, необходимо выполнить следующие действия:

1) внедрить в его тело свой модуль;
2) найти место, которое подлежит коррекции, и осуществить оттуда передачу управления на собственный включенный в тело EXE-файла модуль.

Рассмотрим поэтапно процесс коррекции EXE-файла.

I. Включение в тело исследуемого EXE-файла собственной задачи

На наш взгляд, допустимы следующие варианты включения собственного исполняемого кода в уже существующий EXE-файл:

А. Добавление в конец файла.

Б. Включение в свободное место EXE-файла.
В. Добавление в начало файла.
Г. Включение путем модификации таблицы FAT.
Д. "Склейка" любых EXE-файлов.

Кратко опишем все названные варианты.

А. Добавление в конец файла

Данный вариант очень прост — блок ваших команд добавляется в конец EXE-файла. Только при этом нужно предварительно проверить, возможна ли загрузка ваших команд в оперативную память. Дело в том, что при запуске в работу оверлейного EXE-файла DOS первоначально загружает в память только корневой сегмент. Например, в BORLAND C++ размер BCC.EXE более 1 Мбайта. Ясно, что если в конец файла вы "прилепите" свой модуль, то он просто не попадет в оперативную память и передать на него управление вы не сможете.

Алгоритм включения своих команд в конец EXE-файла представляется следующим.

Проверить по заголовку EXE-файла — какая его часть загрузится в оперативную память, или, иными словами, установить, оверлейный файл или нет. Если в загружаемую часть не входит "хвост" файла, значит данный метод для вас не годится. В противном случае можно совершенно спокойно добавить свой модуль в конец файла и исправить в заголовке размер загружаемой части файла.

Б. Включение в свободное место EXE-файла

В том случае, если вариант А вам не подходит, можно попробовать поискать "пустые" места в загружаемой части EXE-файла, например, область нулей (это скорее всего зарезервированные в программе буферы для размещения данных). Если размер найденного пространства достаточно велик, чтобы разместить в нем ваш модуль, то вам повезло и вы можете включить свои команды в EXE-файл.

В. Добавление в начало файла

Если вам не повезло с вариантами А и Б и вы еще не потеряли надежду довести задуманное до конца, то тогда можно попробовать включить свой модуль в начало EXE-файла, непосредственно за таблицей перемещения, конечно, если там есть место. Ну, а если свободного места нет, то его всегда можно создать — для этого достаточно "раздвинуть" файл и модифицировать таблицу перемещения с учетом сдвига. После данной операции необходимо подправить заголовок, а именно: адрес первой исполняемой команды, размер загружаемой в память области файла, размер заголовка файла.

Г. Включение модуля путем модификации таблицы FAT

Этот вариант предполагает предварительную подготовку вашего модуля в виде обычной исполняемой задачи — COM- или EXE-файла — а затем модификацию только списка распределения кластеров (FAT) без изменения самих исходных файлов.



FAT (File Allocation Table) — это связный список кластеров диска, который DOS использует для отслеживания физического расположения файлов на диске и поиска свободной дисковой памяти для новых файлов. Слово по смещению 1ah в элементе оглавления содержит номер начального кластера для данного файла в списке кластеров FAT. Каждый элемент FAT либо ссылается на следующий кластер, либо указывает на конец цепочки кластеров данного файла.

Для включения собственного модуля в EXE-файл необходимо:

- изменить слово по смещению 1ah в элементе оглавления EXE-файла на номер кластера, начиная с которого на диске расположен ваш модуль;
- по номеру последнего кластера вашего модуля в FAT поставить ссылку на первый кластер EXE-файла;
- пометить в оглавлении включенный в EXE-файл модуль как удаленный.

В результате при запуске EXE-файла на выполнение первоначально отработают ваши команды, после чего вы должны сами позаботиться о загрузке в память и передаче управления исполняемой части EXE-файла, в который вы внедрились.

Д. "Склейка" любых EXE-файлов

Основная сложность при "склейке" двух задач в одну заключается в том, чтобы корректно организовать передачу управления от первой "склеенной" задачи ко второй. В том случае, если первую из задач пишете вы сами, вопрос решается просто: достаточно при завер-



шении задачи передать управление по заранее известным адресам, взятым из заголовка задачи. Если же первая "склеиваемая" задача вам не принадлежит и завершается в неизвестном вам месте и неизвестным вам способом, то в этом случае выход может быть только один — на время работы "склеенной" задачи необходимо доработать часть DOS, ответственную за обработку функции завершения процесса.

Решение данной проблемы возможно двумя способами.

1. Через общий монитор. Создается специальная программа для перехвата всех функций, связанных с завершением работы программ (назовем ее Монитор). Затем "склеиваются" вместе все выбранные EXE- или COM-файлы, но первым в полученном файле размещается Монитор. При запуске задачи Монитор берет на себя обработку всех программных прерываний, по которым может быть завершена задача. Получив управление, Монитор анализирует, какая из задач общего файла была завершена, освобождает ресурсы, самостоятельно выбирает из файла и размещает в памяти следующую задачу, после чего передает на нее управление. И так до тех пор, пока не отработает последняя задача файла. После этого Монитор восстанавливает исходную программу обработки прерываний и завершает работу.

2. Второй вариант является модификацией первого. Разница заключается только в том, что Монитор не обладает суверенитетом отдельной задачи, а прикрепляется к первой "склеиваемой" задаче одним из способов, изложенных в разделе под пунктами А, Б, В, Г. После этого к "зараженной" Монитором задаче можно обычной командой COPY "прилеплять" какие угодно EXE-файлы.

II. Передача управления на включенный в тело EXE-файла модуль

Возможны следующие варианты передачи управления на модуль, внедренный в тело EXE-файла:

А. С помощью изменения начальных значений CS:IP в заголовке EXE-файла.

Б. С помощью изменения первой исполняемой команды по адресу, указанному начальными значениями CS:IP в заголовке EXE-файла.

В. С помощью изменения адресной части команд JMP FAR и CALL FAR в теле файла.

Г. С помощью изменения номера прерывания в двухбайтной команде INT в теле файла.

Кратко рассмотрим названные варианты.

А. Изменение значений CS:IP в заголовке файла

После внедрения блока команд в тело EXE-файла необходимо:

- скопировать содержимое 20-23 байтов заголовка EXE-файла в специальный буфер, предусмотренный для этого во внедренном модуле;
- изменить содержимое 20-23 байтов заголовка EXE-файла на адрес первой исполняемой команды вашего модуля;
- по завершении выполнения вашего блока команд предусмотреть передачу управления по адресу, сохраненному вами в специальном буфере.

Б. Изменение первой исполняемой команды по адресу, указанному начальными значениями CS:IP в заголовке EXE-файла

После внедрения блока команд в тело EXE-файла необходимо:

- скопировать содержимое первых байт по адресу, содержащемуся в 20-23 байтах заголовка EXE-файла, в специальный буфер, предусмотренный для этого во внедренном модуле;
- изменить содержимое первых исполняемых команд EXE-файла (по адресу CS:IP в заголовке EXE-файла) на команду JMP или CALL с адресом внедренного модуля;

- по завершении выполнения вашего блока команд предусмотреть восстановление первых команд EXE-файла и передачу управления по адресу CS:IP в заголовке EXE-файла. Это можно сделать с помощью команд JMP, CALL, RET.

В. Изменение адресной части команд JMP или CALL с атрибутом FAR

Сначала необходимо найти в теле EXE-файла команду JMP FAR или CALL FAR. Это можно сделать простыми средствами без использования дизассемблера. Достаточно найти байт с кодом eah (JMP FAR) или 9ah (CALL FAR) в теле файла и проверить существование элемента в таблице перемещения EXE-файла, указывающего на третий байт после найденного кода. После этого необходимо запомнить адрес перехода (или вызова подпрограммы), откорректировать его в найденном операторе JMP или CALL и предусмотреть передачу управления по первоначальному адресу после отработки всех команд внедренного модуля. Единственная неприятность, которая вас подстерегает при реализации данного метода, — обнаружение команд CALL или JMP, на которые никогда не будет передано управление.

Г. Изменение номера прерывания в двухбайтной команде INT

Этот вариант является самым простым из всех перечисленных.

В данном случае не требуется внедрять ваш модуль в тело корректируемого EXE-файла. Достаточно выполнить его в виде резидентной программы, отрабатывающей какое-либо прерывание по любому свободному вектору, и загрузить ее в память перед запуском EXE-файла.

После этого необходимо исправить в соответствующем месте EXE-файла номер прерывания на тот, для которого вы сделали резидентный обработчик, и запустить файл на выполнение.

Команду INT с соответствующим номером прерывания можно поместить в EXE-файл по адресу первой исполняемой команды или передать управление на обработчик заменой второго байта любой команды INT, находящейся в теле файла. Не забудьте в своей резидентной программе по завершении ее работы восстановить замененный код.

Д. Создание резидентного модуля для обработки прерывания

Этот вариант близок к варианту Г.

В данном случае также не требуется внедрять ваш модуль в тело корректируемого EXE-файла. Достаточно выполнить его в виде резидентной программы, отрабатывающей прерывание, обработку которого вы хотели бы модифицировать. Например, вы забыли сделать в вашей задаче, для которой отсутствуют исходные тексты, проверку корректности вводимых с клавиатуры данных. Самый простой выход здесь — создание собственного резидента для предварительной обработки соответствующей функции прерывания DOS 21h (2, 6, 8). Захватив прерывание 21h, вы дожидаетесь его завершения (организовав передачу управления по завершении работы DOS на себя), проверяете по содержимому стека, из какой части корректируемой программы было выдано прерывание, и выполняете соответствующие действия по обработке данных. После отработки резидента вы либо просто возвращаете управление задаче, либо возвращаете управление с соответствующим кодом завершения. В последнем случае оператору придется повторить ввод данных.

А.Долгин, С.Расторгуев

Использованы материалы:

1. Долгин А.Е. Защита программного обеспечения от исследования. Электронный журнал "НСК", 1992, №1. — М.: Центр СОВМАРКЕТ.
2. Technical Reference. Personal Computer AT. Volume 1.


```
/****************************************************************
* INCL_E.C -- подпрограмма пристыковки исполняемого кода        *
*                 в конец исполняемого модуля.                   *
*                                                                *
* Формат вызова на языке C:                                      *
* INCLUD_E (char far * адрес_буфера_с_исполняемым_кодом,         *
*           unsigned   длина_добавляемого_кода,                  *
*           char far * имя_файла);                               *
*                                                                *
* Возвращает:   1 -- нормальное завершение;                      *
*              -1 -- файл не найден;                             *
*              -2 -- ошибка файловой системы;                    *
*              -3 -- размер COM-файла превышает 64 Кбайт.        *
*                                                                *
* Примечания:                                                    *
* 1) компилятор Borland C++;                                     *
* 2) длина пристыковочной части не должна превышать 64 Кбайт;    *
* 3) длина COM-файла не должна превышать 64 Кбайт;               *
* 4) не осуществляет пристыковку к оверлейным файлам.            *
* 16.05.92.                                                      *
****************************************************************/

#include <dos.h>
#include <mem.h>
#include <stdio.h>
#include <stdlib.h>
#include <io.h>
#include <alloc.h>
#include <fcntl.h>

int includ_e (char far *CodeBuf, unsigned LenBuf,
              char far *FileName)
{
  struct exe_header {  // структура заголовка EXE-файла:
    char signatura[2];         // сигнатура MZ
    unsigned last_page,        // длина последней страницы
             full_page,        // кол-во 512-байтных страниц
             tab_len,          // кол-во эл-тов в табл. перемещ.
             head_len,         // длина заголовка в параграфах
             min_mem,          // мин. и макс. размеры памяти,
             max_mem,          //    распределенной задаче
             start_ss,         // начальные значения SS:SP
             start_sp,
             chk_sum,          // контрольная сумма
             start_ip,         // начальные значения CS:IP
             start_cs,
             beg_tab;          // смещение до табл. перемещения
  } h;

  //    буфер с кодом подготовки стека для передачи управления
  //    на исходную программу для EXE-файлов:
  char buf2[] = { 0x8c,0xd8,              //  mov  ax,ds
                  5,0x10,0,               //  add  ax,10h
                  0x50,                   //  push ax
                  0xb8,0,0,               //  mov  ax,0
                  0x50,                   //  push ax
                  0x8c,0xc8,              //  mov  ax,cs
                  5,2,0,                  //  add  ax,2
```

```
        0x50,                   //  push ax
        0x33,0xc0,              //  xor  ax,ax
        0x50,                   //  push ax
        0xcb,                   //  retf
    0,0,0,0,0,0,0,0,0,0,0,0 };  //  db  12 dup (0) --
                                //  дополнение до 20h

//    буфер с кодом подготовки стека для передачи управления
//    на исходную программу и восстановления первых 3-х байт
//    для COM-файлов:
char buf3[] = { 0xbe,0,1,       //  mov  si,100h
        0xc7,4,0,0,             //  mov  word ptr [si],0
        0xc6,0x44,2,0,          //  mov  byte ptr [si+2],0
        0x1e,                   //  push ds
        0xb8,0,1,               //  mov  ax,100h
        0x50,                   //  push ax
        0x8c,0xc8,              //  mov  ax,cs
        5,0x12,0,               //  add  ax,12h
        0x50,                   //  push ax
        0x33,0xc0,              //  xor  ax,ax
        0x50,                   //  push ax
        0xcb,                   //  retf
        0,0,0,0,0,0 };          //  db  6 dup (0) --
                                //  дополнение до 20h

 char  buf4[] = {0xe9,0,0};       // буфер для jmp на пристыковку
 unsigned i,*p;
 int handle,cc;
 long l;

 if ((handle = _open (FileName, O_RDWR)) == -1)
   return -1;
 if (_read(handle,&h,26) == -1) // считать первые 26 байт файла
 { cc = -2;   goto  exitp; }

 if (h.signatura[0]=='M' && h.signatura[1]=='Z')
 {
 //================== обработка EXE-файла ====================
   movmem (&h.start_ip, buf2+7, 2);
   p = (unsigned *) (buf2 + 3);
   *p += h.start_cs;
   l = lseek (handle, 0L, 2);              // определить длину файла
   _write (handle, &i, 16 - l%16);  // дополнить до параграфа
   _write (handle, buf2, 0x20);
   if(_write (handle, CodeBuf, LenBuf) == -1)  goto  error;
   // определить новые значения CS:IP
   h.start_cs = ((l + 16 - l%16) >> 4) - h.head_len;
   h.start_ip = 0;
   l += 16 - l%16 + 0x20 + LenBuf;   // вычислить новый размер
   h.last_page = l % 512;            // загружаемой части файла
   h.full_page = l / 512;
   if (h.last_page)   ++ h.full_page;
   lseek (handle, 0L, 0);            // записать новый заголовок
   _write (handle, &h, 24);
   cc = 1;  goto  exitp;
 }
 //================== обработка COM-файла ====================
 l = lseek (handle, 0L, 2);                    // длина файла
 if (l > 0xffffL) { cc = -3;  goto  exitp; }
 if (_write (handle, &i, 16 - l%16) == -1)     // дополнить файл
    goto  error;                               // до параграфа
 l += 16 - l%16;
 p = (unsigned *) (buf3 + 19);
 *p += (unsigned) (l >> 4);
 p = (unsigned *) (buf4 + 1);
 *p = (unsigned) (l - 3);
 lseek (handle, 0L, 0);              // записать JMP на пристыковку
 _write (handle, buf4, 3);
 movmem (&h, buf3+5, 2);
 buf3[10] = (char) h.last_page;
 lseek (handle, 0L, 2);
 if (_write (handle, buf3, 0x20) == -1)        // записать buf3
    goto  error;
 if (_write (handle, CodeBuf, LenBuf) == -1)   // записать
    goto  error;                               //   пристыковку
 cc = 1;

exitp:
 _close (handle);
 return cc;

error:
 cc = -2;
 goto  exitp;
}
```



Разработчики защит хорошо знают, что искусственно испорченные сектора ключевой дискеты можно запросто эмулировать, написав собственный обработчик прерывания 13h. Однако не все знают, что искусный хакер может добиться той же цели и гораздо более изощренным способом, обманув защиту так, что "комар носа не подточит".

Эмуляция сбойных секторов

"Большинство людей не хочет плавать, до того как научится плавать, и, конечно, они не хотят думать, ведь они рождены, чтобы жить, а не думать."

Новалис

Одним из способов обхода защиты, основанной на искусственных дефектах дистрибутивного носителя (лазерные метки и т.п.), может быть написание программного эмулятора особенностей дискеты [1]. Такой эмулятор перехватывает прерывание 13h и анализирует каждое обращение к дискете. Если защищенная программа предпринимает попытку записи/чтения секторов, которые были физически испорчены на оригинальном дистрибутивном носителе, то эмулятор возвращает управление в программу, выставив флаг ошибки.

Исключить программную эмуляцию особенностей носителя можно только одним способом — напрямую обращаться из защищенной программы к контроллеру дисководов. Как ни странно, находятся программисты, которые то ли из боязни получить не 100%-совместимую программу, то ли от неумения работать с аппаратурой напрямую, пренебрегают этим приемом. Хотя им известно о возможности написания эмулятора, они уповают на то, что им каким-то образом удастся узнать "истинный" адрес прерывания 13h и обмануть эмулятор, передавая управление обработчику этого прерывания напрямую с помощью команды jmp.

Я не берусь рассуждать о том, можно или нет документированными способами узнать адрес обработчика прерывания 13h в BIOS. Эта тема заслуживает отдельного разговора. Тем не менее, существует возможность обхода защиты и с передачей управления прерыванию 13h напрямую. Эта возможность, о которой почему-то забывают, а может и не подозревают авторы подобных "защит", основана на обработке аппаратного прерывания IRQ 6. Сразу хочу обратить внимание неспециалистов. IRQ 6 — это номер аппаратного прерывания, и какой ему будет соответствовать вектор прерывания, зависит от вашей машины. Например, IBM AT располагает обработчик этого прерывания по вектору 08h, на моей Crystal Data AT он располагается по вектору 0Eh. В дальнейшем, чтобы не возникало разночтений, обработчик, на который указывает соответствующий вектор прерывания, будем просто называть "обработчик IRQ 6".

Единственная задача, которую выполняет стандартный обработчик IRQ 6, заключается в выставлении в области данных BIOS флага, означающего окончание дисковой операции [2]. Если обработчик сделать более "интеллектуальным", то он с успехом может заменить эмулятор на векторе 13h. Естественно, возникают два вопроса: на чем основывается эта идея и как ее реализовать.

В основу идеи "интеллектуального" обработчика IRQ 6 положены следующие особенности работы контроллера дисководов. Во-первых, по завершении любых дисковых операций контроллер дисковода возвращает от одного до семи байт статуса завершения последней дисковой операции. Причем для операций чтения/записи байты статуса завершения содержат координаты сектора, к которому производилось обращение [2]. Это свойство контроллера может быть использовано для селекции секторов на "хорошие" и "плохие". Во-вторых, если в поле статуса последней операции контроллера и в поле статуса последней

дисковой операции выставить соответствующие флаги ошибки (см. таблицу), то обработчик прерывания 13h, к которому попадет управление после обработки IRQ 6, сам сообщит запрашивающей программе о неудаче при доступе к сектору.

Вся сложность реализации эмулятора заключается в том, что контроллер может выдать байты статуса только один раз. В свою очередь, обработчик прерывания 13h "знает", что по завершении дисковой операции должен быть возвращен по крайней мере один байт статуса. Следовательно, если эмулятор уже считал байты статуса, то обработчик прерывания 13h, обнаружив, что контроллеру дисководов уже нечего ему "сказать", выдаст сообщение об ошибке. Такой вариант нас не устраивает, поскольку эмулятор должен пропускать обращения к обычным секторам.

Поля в области данных BIOS, используемые программой — эмулятором дефектов дискеты

Адрес	Размер	Значение поля
40:3F	1 байт	Статус мотора НГМД бит 7 идет запись 3 мотор вкл. (НГМД 3) 6 2 мотор вкл. (НГМД 2) 5 1 мотор вкл. (НГМД 1 -- B:) 4 0 мотор вкл. (НГМД 0 -- A:)
40:41	1 байт	Статус последней дисковой операции НГМД бит 7 таймаут 3 ошибка DMA 6 ошибка поиска 2 сектор не найден 5 ошибка контр. 1 адрес не найден 4 ошибка CRC 0 неверная команда
40:42	7 байт	Статус последней операции контроллера НГМД
40:94	2 байта	Текущая дорожка для устройств 0 и 1

Анализ сочетания особенностей работы контроллера дисководов и обработчика прерывания 13h приводит к выводу, что для нормальной работы эмулятора он должен обманывать еще и BIOS. Делать он это может следующим образом. Байты статуса завершения последней дисковой операции считываются на свое законное место — по адресу 40:42 (см. таблицу). После этого независимо от того, был ли сектор помечен как сбойный или нет, контроллеру посылается команда "дай статус контроллера без генерации прерывания". В этом случае после возврата управления обработчику прерывания 13h контроллер выдаст ему единственный байт статуса, который будет положен на место первого байта статуса завершения последней операции чтения/записи сектора. При этом все остальные байты статуса завершения операции чтения/записи останутся нетронутыми.

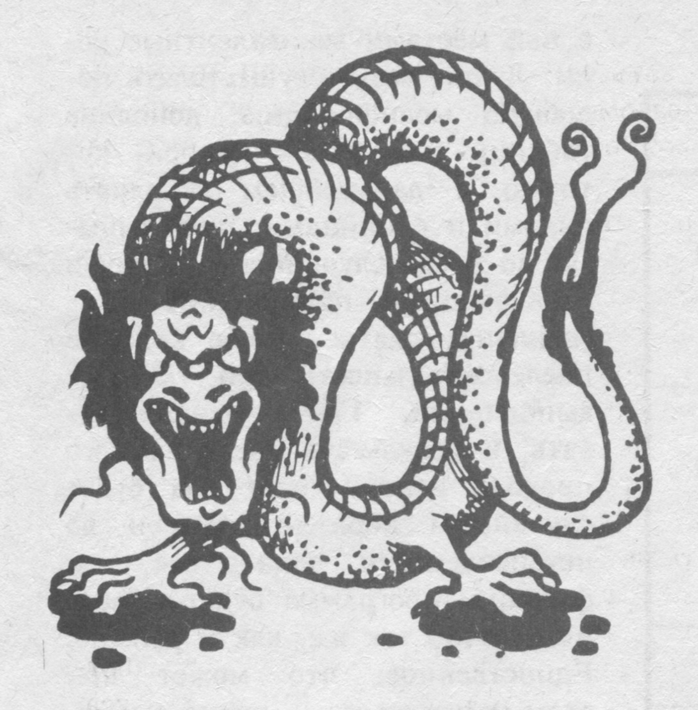
Таким образом мы получаем желаемый результат: с одной стороны — эмулятор определил, к какому сектору был осуществлен доступ, а с другой — обработчик прерывания 13h получил ожидаемый байт статуса.

По сути дела, в данной ситуации игра идет на том, что обработчик прерывания 13h не совсем досконально разбирается с возвращаемыми байтами статуса [3]: первый байт статуса после операции чтения/записи сектора и байт статуса контроллера имеют различные значения. Но тем не менее, обработчик прерывания 13h допускает такую замену

Д.Правиков

Использованы материалы:

1. Расторгуев С.П., Дмитриевский Н.Н. Искусство защиты и "раздевания" программ. — М.: Центр СОВМАРКЕТ, 1991. — С. 27.
2. Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT. — М.: Финансы и статистика, 1992. — С. 339 — 351.
3. Technical Reference. Personal Computer AT. Volume 1.



Вы хотите запутать алгоритм своей программы так, чтобы в нем никто кроме вас не мог разобраться? Тогда, возможно, предлагаемый подход динамического изменения кода исполняемого модуля — для вас.

Самомодифицирующиеся программные модули

"Никогда не говори "слава богу", избавившись от маленькой неприятности, ибо неизвестно, не ожидает ли тебя неприятность крупная."

Восточная мудрость

Актуальность защиты от НСК обрабатываемых на ПЭВМ данных не вызывает сомнений. На сегодняшний день в этой области создано немало интересных разработок и специальных алгоритмов шифрации данных. Особый интерес вызывают динамически изменяемые алгоритмы шифрации.

При обработке информации на ПЭВМ алгоритмы шифрации реализуются в виде обычных программ. На наш взгляд, именно программа, исполняемый модуль становится самым слабым звеном при попытке НСД к шифрованным данным. Дизассемблирование или исследование задачи под отладчиком часто позволяет разобраться с алгоритмом шифрации и получить доступ к "закрытой" информации. Проблеме защиты программного обеспечения от исследования отладчиком была посвящена наша статья "Защита программ от дизассемблеров и отладчиков" (КомпьютерПресс №4´92) и отчасти статья "Железные изюминки защиты" (КомпьютерПресс №8´92).

В данной работе рассматривается принципиально иной подход к защите ПО от исследований, базирующийся на динамическом изменении исполняемого кода и самого исходного EXE- или COM-файла. Полное изменение исполняемого кода становится возможным благодаря адекватной замене одной или нескольких команд программы другой последовательностью команд без изменения выполняемых программой функций. Например, команда MOV может быть заменена парой команд PUSH и POP, команда CALL — парой PUSH и JMP и т.д. Перечень взаимозаменяемых кодовых последовательностей, охватывающий некоторые основные команды ассемблера, приведен в таблице.

Из сказанного следует, что всегда можно организовать работу программы так, чтобы при каждом ее выполнении происходила замена исполняемых команд на эквивалентные (согласно таблице). При этом необходимо постоянно изменять и саму таблицу, переставляя



Взаимозаменяемые кодовые последовательности для основных команд ассемблера

Команды пересылки	
mov op1,op2	push op2 pop op1
xchg op1,op2	push op1 mov op1,op2 pop op2
	push op1 push op2 pop op1 pop op2
lds r,dword ptr op	mov r,word ptr op mov ds,word ptr op+2
les r,dword ptr op	mov r,word ptr op mov es,word ptr op+2
Арифметические команды	
add op1,op2	push ax mov ax,op2 add op1,ax pop ax
	xchg op2,ax add op1,ax xchg op2,ax
adc op1,op2 sub op1,op2 sbb op1,op2 и др.	аналогично add
inc op	add op,1
dec op	sub op,1
Логические команды	
and op1,op2 or op1,op2 xor op1,op2 и др.	аналогично add
not op	xor op,0ff(ff)h

Цепочечные команды	
rep movsb	push ax m: mov al,[si] mov es:[di],al inc si inc di loop m pop ax
repe(repne) cmpsb	push ax m: mov al,[si] cmp al,es:[di] jne(je) m1 inc si inc di loop m m1: pop ax
lodsb	mov al,[si] inc si
stosb	mov es:[di],al inc di
shift op,cnt	push cx mov cx,cnt m: shift op,1 loop m pop cx
Команды передачи управления	
j(условие) loc	jn(условие) loc1 jmp loc loc1: . . .
loop loc	dec cx jne loc
jmp addr	push addr ret
jmp far addr	push addr+2 push addr retf
call addr	push m jmp addr m: . . .

Принятые сокращения:
op,op1,op2 -- операнды команд; cnt -- счетчик в командах сдвига;
r -- операнд-регистр; loc -- метка в командах перехода и цикла;
shift -- код команды сдвига; addr -- адрес в командах перехода и вызова п/программ.



в ней местами эквивалентные коды. Лучше всего осуществлять подобную модификацию, дополнив таблицу еще одной графой, которую в дальнейшем заполнять нулями и единицами, полученными по псевдослучайному закону и означающими, например: 0 — не модифицировать данную кодовую последовательность при текущем выполнении, 1 — модифицировать. В результате после каждого прохода исполняемый код будет случайным образом изменен до неузнаваемости, однако все свои функции программа будет выполнять точно так же, как и раньше. Единственное, что может при этом измениться, — время выполнения программы.

В первой графе представленной таблицы записано мнемоническое обозначение исходного оператора, адекватная интерпретация которого приведена во второй графе. В общем случае количество альтернатив для исходного кода может быть более одной — это целиком определяется вашими потребностями и зависит от фантазии.

Представленная таблица имеет общий вид и вряд ли может быть применена на практике. Поэтому для каждой программы следует создавать индивидуальную таблицу взаимозаменяемых кодов, которые в качестве операндов команд могут иметь конкретные часто встречающиеся в программе значения. При этом для выравнивания длин кодовых последовательностей можно использовать команду NOP или пару PUSH, POP. Эти же команды можно использовать для получения взаимозаменяемой последовательности кодов в тех случаях, когда для команды нет альтернативы или она слишком длинна. Например, NOP, CLI можно заменить на CLI, NOP, и т.п.

Практически реализовать изложенный подход к самомодификации исполняемых модулей существенно проще для процессоров типа 80286 и выше в силу того, что они имеют более обширный набор команд. Кроме того, отдельным кодовым последовательностям может быть поставлена в соответствие не одна, а несколько функционально эквивалентных кодовых последовательностей.

В заключение мы приводим пример самомодифицирующейся программы.

А.Долгин, С.Расторгуев

```
;==============================================================;
;          Пример самомодифицирующейся программы               ;
;==============================================================;

code segment para public
        assume cs:code,ds:code
        org  100h
main    proc
start:
        jmp  begin

vec21   dw   21h*4,0                 ; указатель на в.п. 21h
old21   dw   ?,?                     ; указатель на обработчик 21h

;---------- структура таблицы взаимозаменяемых команд ----------
; длина команды ! первичный код ! вторичный код ! список адресов ! 0ffffh

tabcod  dw   offset cmd1_2-offset cmd1_1  ; длина команды 1
cmd1_1: pushf                        ; первичный код
        call dword ptr old21
cmd1_2: nop                          ; вторичный код
        int  21h
        nop
        nop
        dw   offset adr1_1           ; список адресов
        dw   offset adr1_2
        dw   offset adr1_3
        dw   offset adr1_4
        dw   offset adr1_5
        dw   offset adr1_6
        dw   0ffffh                  ; конец списка адресов

        dw   offset cmd2_2-offset cmd2_1  ; длина команды 2
cmd2_1: inc  bx                      ; первичный код
        inc  bx
        nop
cmd2_2: add  bx,2                    ; вторичный код
        dw   offset adr2_1           ; список адресов
        dw   offset adr2_2
        dw   0ffffh                  ; конец списка адресов
; . . . . . . . . . . . . . . . . . . . . . . . . . . . .

        dw   0ffffh                  ; конец таблицы TABCOD
begin:
        push cs
        pop  ds
        mov  ah,9                    ; вывод сообщения
        mov  dx,offset msg

adr1_1: nop
        int  21h
        nop
        nop
        call modif                   ; модифицировать программу
        mov  ax,3d02h                ; открыть файл программы
        mov  dx,offset filename

adr1_2: nop
        int  21h
        nop
        nop
        jc   error
        mov  bx,ax                   ; записать в него
        mov  ah,40h                  ;  модифицированную программу
        mov  cx,offset fin - offset start
        mov  dx,offset start
adr1_3: nop
        int  21h
        nop
        nop
        jc   error

exit:                                ; закрыть файл программы
        mov  ah,3eh

adr1_4: nop
        int  21h
        nop
        nop
        mov  ax,4c00h                ; завершить задачу
adr1_5: nop
        int  21h
        nop
        nop

error:
        mov  ah,9                    ; вывод сообщения об ошибке
        mov  dx,offset msg_err
adr1_6: nop
        int  21h
        nop
        nop
        jmp  short exit

main    endp

;----------------------------------------------------------------
; Подпрограмма модификации исполняемого файла в соответствии
; с таблицей взаимозаменяемых команд

modif   proc near

        cld
        les  bx,dword ptr vec21      ; сохранить адрес
        les  bx,dword ptr es:[bx]    ;  обработчика 21h
        mov  old21,bx
        mov  old21+2,es
        push cs
        pop  es
        mov  si,offset tabcod
rpt1:
        mov  cx,[si]                 ; длина команды
        add  si,2
        mov  bx,cx                   ; получить в BX адрес списка
        shl  bx,1                    ;  адресов для текущей команды
        add  bx,si
rpt2:
        cmp  [bx],0ffffh             ; конец списка адресов?
        je   nxt_modif               ; да -- nxt_modif
        mov  di,[bx]                 ; нет -- переслать первичный код
adr2_1: add  bx,2                    ;  по адресу [bx]
        push cx
        rep  movsb
        pop  cx
        sub  si,cx                   ; перейти к следующему адресу
        jmp  short rpt2
nxt_modif:
        mov  di,si                   ; поменять в таблице местами первичный
        add  di,cx                   ;  и вторичный коды данной команды
rpt3:
        mov  al,[si]
        xchg al,[di]
        mov  [si],al
        inc  si
        inc  di
        loop rpt3
adr2_2: add  bx,2                    ; перейти к следующей команде
        cmp  [bx],0ffffh             ; конец таблицы?
        je   end_modif               ; да -- end_modif
        mov  si,bx
        jmp  short rpt1
end_modif:
        ret

msg      db  ' Hello, World !', 0ah,0dh,'$'
filename db  'hello.com',0
msg_err  db  'Ошибка работы с файлом!', 0ah,0dh,'$'

modif   endp

fin:

code    ends
        end  start
```

Как скрытно пометить файл

"Небо знает, земля знает, ты знаешь, я знаю — кто говорит, что никто не знает?"

Восточная пословица

Существует очень простой способ пометить файл таким образом, чтобы сделанную вами отметку нельзя было обнаружить привычными средствами типа XTGOLD или NC, а также чтобы она не копировалась вместе с файлом при его копировании традиционными средствами, в частности, командой COPY.

В основу данного способа положен тот факт, что при любом чтении данных с магнитного носителя (дискета, винчестер) минимальная порция считываемых данных кратна размеру сектора. Даже в случае, если ваш файл состоит всего из одного байта, при чтении этого байта с диска или с дискеты реально будет считано минимум 512 байт. При этом непосредственно в вашу задачу будет передан только один байт, который DOS выберет из так называемого дискового буфера. Но в самом дисковом буфере будет находиться весь считанный сектор.

Средствами практически любого языка программирования, например Turbo C, очень легко можно скрытно пометить файл, помещая нужные коды за границу файла, а потом прочитать эту метку. При этом, в случае копирования файла обычными средствами, метка копироваться не будет.

Пример 1

```
/* Запись метки за границу файла, имя которого
   задается в командной строке запуска задачи. */

#include <stdlib.h>
#include <io.h>

main (int ig, char *v[])
{
  unsigned char a[] = "label";   /* метка из 5 символов */
  int n, j, k;
  long l;

  n = _open (v[1], 4);         /* открыть файл */
  if (n < 0)   abort ();
  l = lseek (n, 0L, 2);        /* определить его длину в байтах */
  _write (n, a, 5);            /* записать в конец файла метку */
  _close (n);                  /* закрыть файл */

  n = _open (v[1], 4);         /* снова открыть файл */
  if (n < 0)   abort ();
  chsize (n, l);               /* уменьшить его размер на 5 байт */
  close (n);                   /* закрыть файл */

  exit (0);
}
```

Пример 2

```
/* Чтение метки за границей файла.
   Имя файла задается в командной строке запуска задачи. */

#include <stdlib.h>
#include <io.h>
#include <stdio.h>
#include <string.h>

main (int ig, char *v[])
{
  unsigned char a[6];
  int n;
  long l;

  n = _open (v[1], 4);          /* открыть файл */
  if (n < 0)   abort ();
  l = lseek (n, 0L, 2);         /* определить длину файла */
  lseek (n, 5L, 2);             /* уйти за границу файла на 5 байт */
  _write (n, a, 0);             /* зафиксировать новую длину файла */
  _close (n);                   /* закрыть файл */

  n = _open (v[1], 4);          /* открыть файл заново */
  if (n < 0)   abort ();
  /* Установить указатель на 5-й от конца файла байт.
     На данный момент в файл уже включена метка -- это
     последние 5 байт файла */
  lseek (n, -5L, 2);
  _read (n, a, 5);             /* чтение метки */
  chsize (n, l);               /* восстановить исходный размер файла */
  close (n);                   /* закрыть файл */

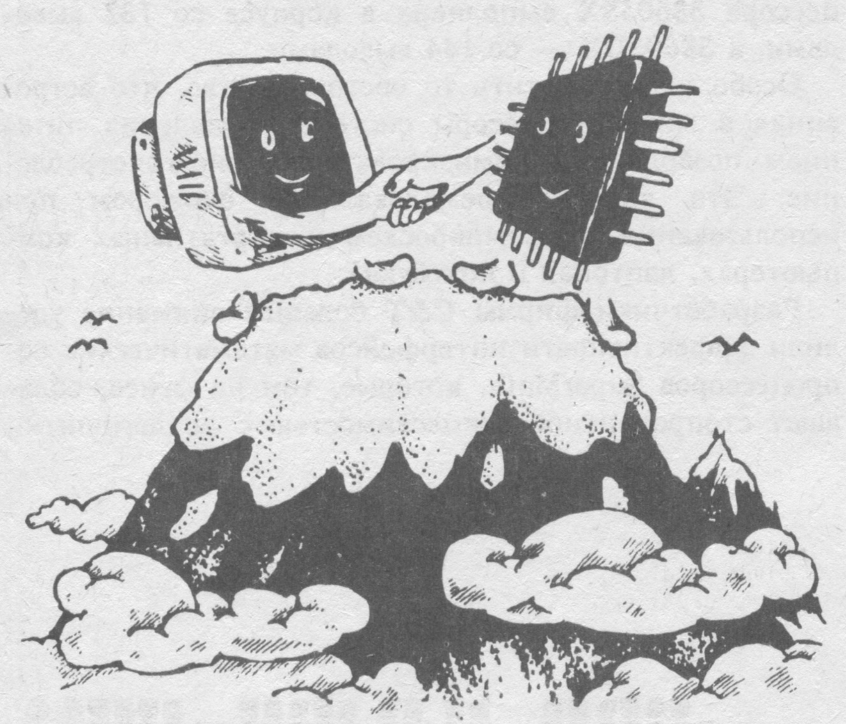
  if (!strncmp (a, "label", 5))
    printf("\nМетка обнаружена.");
  else
    printf("\nМетка отсутствует!");
  exit (0);
}
```

Остается только добавить, что, записывая метку "label" за границу файла, мы предположили, что за его концом достаточно дискового пространства до границы кластера. Для корректности примера 1 соответствующую проверку предлагаем вам написать самостоятельно.

А.Долгин, С.Расторгуев

Тел. (095) 522-61-11

В своих статьях мы попытались донести до читателей часть результатов, полученных в процессе исследований. Прекрасно понимаем, что это лишь малая толика великого множества возможных решений. Пространство алгоритмов и разнообразных приемов программирования безбрежно, а пытающихся зачерпнуть из него ложкой можно охарактеризовать словами неизвестного восточного мудреца: "Сколько муравей ни ползает по железному столбу, тот все равно не износится".



Работая с Summit Systems, вы работаете с CHIPS&Technologies!

Сегодня купить персональный компьютер — дело совсем не хитрое. Газеты и журналы пестрят не только полосными рекламами, подробно рассказывающими о технических характеристиках предлагаемых изделий, но и "телеграфными" объявлениями типа: "AT/386-33, 4-120, супер, косые, широкий, мышь". Предложение сейчас, как говорится, превышает спрос. Нередко поэтому потенциальные покупатели оказываются в роли Буриданова осла, который, как известно, при отсутствии какого-либо мотива к предпочтению одной охапки сена другой умер бы с голоду. И хотя для покупателя мотивов при выборе подходящей техники существует немало, наиболее весомым (да и распространенным) по-прежнему остается один — стоимость компьютера. Большинство почему-то до сих пор предпочитает оптимизировать задачу покупки только по этому параметру. Трудно, конечно, не согласиться, что цена — это условие необходимое, но, увы, не достаточное.

Справедливости ради надо отметить, что горький опыт массовой "персонализации" все же кое-чему научил наших пользователей. Теперь после вопроса о цене на оборудование можно услышать вопрос и о его качестве. На это, кстати, многочисленные продавцы вычислительной техники дружно откликнулись в своих рекламных объявлениях строчкой "1 год гарантии".

Тем, кто так или иначе имел (или имеет) дело с сервисным обслуживанием электронных устройств, известно, что один год гарантийной работы оборудования — срок небольшой (хотя и общепринятый). Фирма, которая декларирует более длительную безотказную работу своих компьютеров, должна, как говорится, "на все сто" быть уверена в качестве своих изделий. Понятно, что найти такие компьютеры нелегко. Но можно. Если вы обратитесь, например, на фирму Summit Systems.

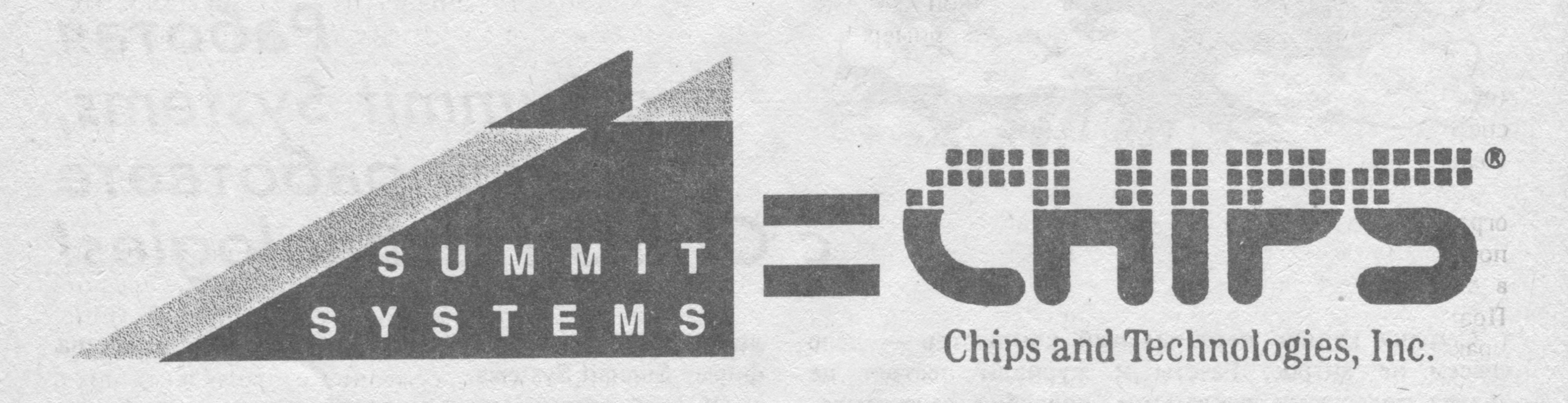
Об этой сравнительно молодой, но уже достаточно известной фирме и ее достаточно хорошо зарекомендовавших себя компьютерах написано (и сказано) немало добрых слов. Но сегодня мы поговорим не о самих компьютерах, а об их полупроводниковой "начинке", которая позволяет фирме Summit Systems создавать не только элегантные, но и высоконадежные и производительные системы (гарантийный срок компьютеров Summit Systems составляет 2 года!).

Как известно, на системных платах компьютеров с маркой Summit Systems используются наборы микросхем фирмы Chips&Technologies (C&T). Эта американская фирма по праву входит в тройку крупнейших производителей микросхем для персональных компьютеров. Если при создании Summit Systems фирма C&T являлась просто одним из партнеров, то теперь Summit Systems не только представляет ее интересы, но и является, по сути, подразделением C&T на территории СНГ. Так что, если говорить о продукции Summit System, то можно смело подразумевать и C&T.

Недавно мы уже рассказали нашему читателю о замечательных свойствах наборов микросхем NEAT, выпускаемых C&T (КомпьютерПресс №7'92), но ассортимент продукции фирмы ими, конечно, не исчерпывается. Так что продолжим разговор о других (смею заверить, не менее интересных) микросхемах фирмы C&T, не забывая при этом, что их достоинства — это потенциальные достоинства компьютеров Summit Systems. Ведь именно использование самой последней американской технологии позволяет вам (с помощью этих компьютеров, разумеется) заглянуть в свое будущее...

Новые микросхемы "от C&T"

В соответствии со своей стратегией разработки совместимых микропроцессоров фирма C&T создала целый ряд микросхем, которые по своим возможностям не только сравнимы, но и превосходят оригинальные продукты, такие как микропроцессоры i386DX/SX и математические сопроцессоры i387DX/SX. Кроме того, созданные микросхемы могут быстрее работать с обычными программами, потребляя при этом меньше мощности. Набор этих микросхем носит название Super386 ChipSystem Architecture и состоит из четырех



32-разрядных микропроцессоров (38600DX/SX, 38605DX/SX), двух математических сопроцессоров (SuperMath DX/SX) и однокристального компьютера PC/Chip (F8680, all-in-line).

Первые два микропроцессора (38600DX/SX) полностью совместимы, в том числе и по выводным контактам, с соответствующими микропроцессорами фирмы Intel, но работают производительнее последних приблизительно на 10%. Другие два 32-разрядных микропроцессора — 38605SX и 38605DX — дополнительно включают в себя кэш для инструкций емкостью 512 байт, что позволяет им работать уже на 40% производительнее, чем (эквивалентные по тактовой частоте) предшествующие модели 386-х микропроцессоров. Например, при использовании микропроцессора 38605DX с тактовой частотой 40 МГц на системной плате с набором NEAT PEAKSetDM гарантируется производительность не менее 11 MIPS. Кстати, оба эти микропроцессора (как и микросхема F8680) включают специальный режим SuperState, о котором мы расскажем несколько позже. Другой особенностью каждого такого микропроцессора является специальный вывод управления (isolate control pin), который позволяет переводить все выводы в так называемое "третье" (высокоимпедансное) состояние и запретить все вводные сигналы. Возможность чтения всех внутренних регистров микропроцессоров существенно улучшает операции тестирования и проверки системы.

Введение кэша во внутреннюю структуру микропроцессоров вынудило разработчиков увеличить количество выводов для этих микросхем. Версия микропроцессора 38605SX выполнена в корпусе со 132 выводами, а 38605DX — со 144 выводами.

Особо надо отметить то обстоятельство, что встроенная в микропроцессоры система управления питанием позволяет минимизировать их энергопотребление. Это является немаловажным фактором при использовании этих микросхем в портативных компьютерах, лэптопах и ноутбуках.

Разработчики фирмы C&T большое внимание уделили эффективности интерфейсов математических сопроцессоров SuperMath, которые, тем не менее, обладают стопроцентной совместимостью с аналогичными микросхемами фирмы Intel. Главным усовершенствованием SuperMath является модернизация их внутренней структуры. Это позволяет им при выполнении некоторых операций работать в 6 раз производительнее оригинальных математических сопроцессоров i80387. Например, простая операция типа стек-стек требует только трех тактов работы, а довольно сложная операция извлечения квадратного корня из числа с плавающей запятой — 19 тактов. Оригинальным математическим сопроцессорам для выполнения этих операций необходимо 11 и 122 такта соответственно. Версии математических сопроцессоров SX могут работать на частотах 16, 20 и 25 МГц, а версии DX — на частотах 33 и 40 МГц.

Что такое SuperState?

Известно, что современная компьютерная индустрия строится на стандартах. Однако понятно, что приверженность стандартам создает и определенные ограничения на имеющиеся возможности для разработчиков новых систем. Стандартная архитектура аппаратных средств персонального компьютера предполагает, например, что машина имеет конфигурацию, включающую такие периферийные устройства, как клавиатура, экран, один или несколько жестких дисков и т.п. Предполагается также, что эти периферийные устройства взаимодействуют с процессором строго определенными, устоявшимися, хотя иногда и устаревшими способами. Таким образом, эта ограниченная

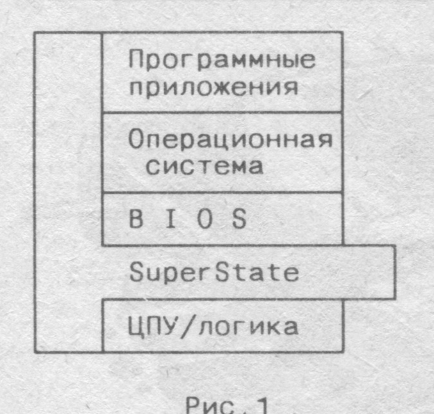


Рис. 1

стандартная архитектура создает трудности при создании новейших видов персональных компьютеров, таких как лэптопы и ноутбуки, которые могут использовать новые типы устройств, никогда не применявшиеся для настольных компьютеров.

Попытка создать стандартное программное обеспечение для персонального компьютера предполагает, что приложение полностью подчиняется имеющейся аппаратной системе. Это допущение усложняет разработку, например, таких важнейших функций, как управление питанием (особенно для портативных систем) или добавление к системе устройств, работающих в реальном времени (real time device).

Хотя жесткое следование стандартам на практике и ограничивает возможности разработчиков, они все же понимают, что расширение возможностей, достигаемое в обход этих стандартов, обходится слишком дорого. Поэтому разработчики многих фирм до сих пор ищут практические способы приблизить свои продукты к нуждам специфических групп пользователей или приложений.

Специалисты фирмы C&T взялись за решение этих проблем, расширив архитектуру оригинальных "интеловских" микропроцессоров 80x86, добавив новый режим SuperState. Этот операционный режим гарантирует, по сути, прозрачный интерфейс между программными и аппаратными средствами и поддерживается в микросхемах PC/Chip и микропроцессорах 38605DX/SX. Режим SuperState обеспечивает некое резервное пространство в системе, что позволяет работать с новейшим программным обеспечением. SuperState также обеспечивает изменение старых и введение новых функций в системе. Однако этот режим не использует основную память или регистры и является, вообще говоря, неощутимым для программных приложений, операционной системы и BIOS.

Для разработчиков системы SuperState является отличным средством для реализации различных новшеств. Системные функции могут быть изменены почти произвольным образом так, что аппаратная часть будет совместима с любыми операционными системами.

В среде SuperState связь программных средств анализируется стандартной программой, написанной в так называемом SuperState-коде. Эта программа проверяет на совместимость и аппаратные средства. Если нужно, комбинация режима и кода SuperState "виртуализирует" команду и переводит ее в такую форму, которую "понимает" аппаратура. Код SuperState состоит из инструкций для семейства процессоров 80x86, причем выполняются эти инструкции, конечно, только в режиме SuperState. Стандартные программы, написанные с учетом выполнения этих инструкций, обеспечивают специализированные функции, которые управляют операциями, ожидаемыми в режиме SuperState.

В связи с этим процесс виртуализации может использоваться в одном случае для перехвата операций ввода-вывода и прерываний или, если это необходимо, для проверки или эмулирования совместимости, чтобы затем разблокировать прерывание и продолжить цикл. В другом случае этот процесс может быть использован для поддержки системы управления энергопотреблением, которая приводит в действие процессы Suspend и Resume. Режим SuperState может обеспечить функционирование двух несовместимых операционных систем одновременно или добавить необходимые возможности для повышения производительности специальных прикладных программ, входящих в прикладное программное обеспечение.

Например, если драйвер программного обеспечения написан непосредственно для конкретного устройства "в обход" операционной системы, он, безусловно, сложен для модификации или перехвата операций ввода-вывода, но только до тех пор, пока программист не получит доступа к режиму SuperState. Если процессор обнаруживает передачу, он переключается в режим SuperState и дает возможность устройству управления завершить операцию. Среда режима SuperState отделена от нормального операционного режима, который включает DOS и BIOS. Будучи "визуализирована", эта среда могла бы появиться между системной логикой микропроцессора и BIOS; во время же обычных операций процессора она могла бы быть "прозрачной" для всех инструкций (рис. 1). Функции SuperState управляют практически всей деятельностью системы (прерывания, операции ввода-вывода и т.д.) и таким образом могут эмулировать соответствующую среду независимо от совместимости. Надо отметить, что процессор имеет несколько возможностей (или условий), при которых он может входить в режим Super-

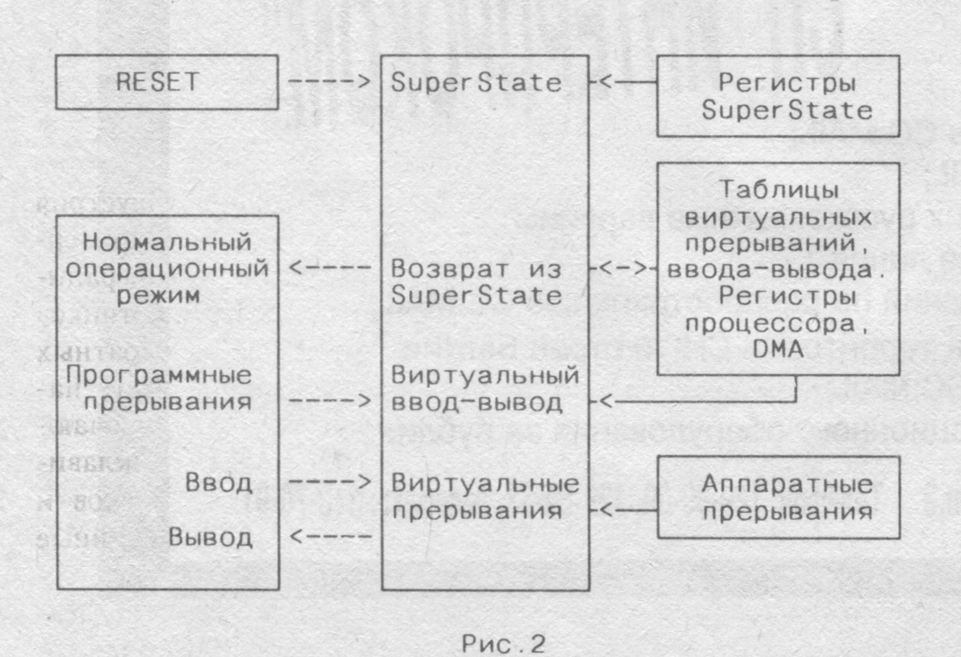
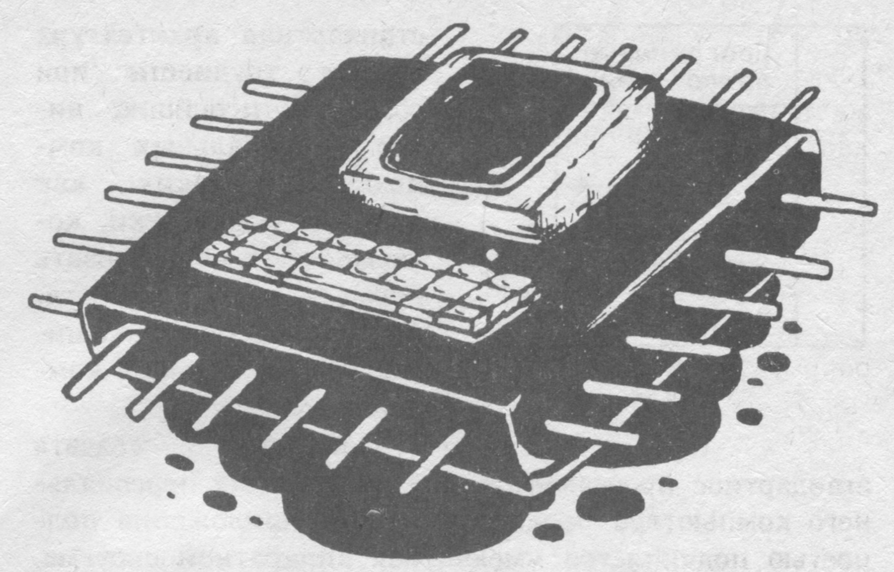


Рис. 2

State. Это происходит, например, при установке микросхемы в исходное состояние (reset), при перехвате прерываний программных или аппаратных средств, при исполнении команды ввода-вывода, когда необходимо эмулировать или осуществить управление устройством на шине ввода-вывода. Другие условия могут заключаться в том, что специально отведенное время для выполнения задачи истекло, или канал прямого доступа (DMA) должен быть инициализирован автоматически, или необходимо обслужить внешний запрос, который был активирован на одном из программируемых выводов микропроцессора. На рис. 2 схематично показано, каким образом система сможет получить доступ к возможностям SuperState.

Частью режима SuperState является также возможность устанавливать параметры конфигурации системы (setup). Как известно, прикладное программное обеспечение не может оказывать на эту информацию никакого существенного влияния. При использовании режима SuperState можно перехватывать и интерпретировать формат начальной конфигурации, так что прежний setup и код BIOS могут быть использованы без изменений. Различные аппаратные новшества в архитектуре системы, которые включают, например,



карты флэш-памяти или другие нестандартные устройства, могут легко управляться в режиме SuperState и использоваться поэтому даже с текущей версией DOS. Так как режим SuperState автоматически управляет работой микропроцессора, программист освобождается от таких отвлекающих операций, как, например, управление работой системы ввода-вывода. Режим SuperState может быть использован также во время отладки программ. Управление буфером трассиров-

ки позволяет программисту отслеживать ошибки из точки прерывания программы, что существенно снижает необходимость в тестовых устройствах, таких, например, как встроенные эмуляторы.

PC в одной микросхеме

Для большинства начинающих пользователей микропроцессор отождествляется с компьютером. Хотя микропроцессор действительно предоставляет важнейшие вычислительные функции и зачастую определяет потенциальную мощность любого персонального компьютера, однако полные системы PC включают в себя множество функций, которые не обеспечиваются их процессорами.

Графические функции компьютера, возможность хранения информации на магнитных носителях, память, средства коммуникации представлены обычно собственными подсистемами. Вся совокупность этих подсистем, составляющих конфигурацию системы, и определяет действительную производительность системы. Большинство подсистем выполнены в виде отдельных плат или микросхем, нуждающихся в свою очередь в других микросхемах поддержки.

Первоначальные IBM PC содержали на системной плате большое количество отдельных микросхем. В настоящее время получили развитие стандартные наборы микросхем (chip set), которые выполняют практически все функции подсистем. Использование таких наборов резко снизило общее количество микросхем, необходимых для создания системы и в конечном итоге способствовало снижению стоимости и сложности разработки, свойственных созданию клона PC. Немаловажным фактором использования наборов микросхем является и повышение надежности создаваемой системы.

Понятно, что если объединить микропроцессор и набор микросхем в единую микросхему, то затраты на создание персонального компьютера можно снизить еще более существенно. К тому же в этом случае возможен общий рост производительности и надежности разрабатываемой системы. Представляя себе все значительные выгоды этого начинания, многие фирмы — производители полупроводниковой техники — начали разработку "PC в одной микросхеме". Первой этой цели добилась фирма C&T, недавно анонсировавшая свою микросхему PC/Chip F8680.

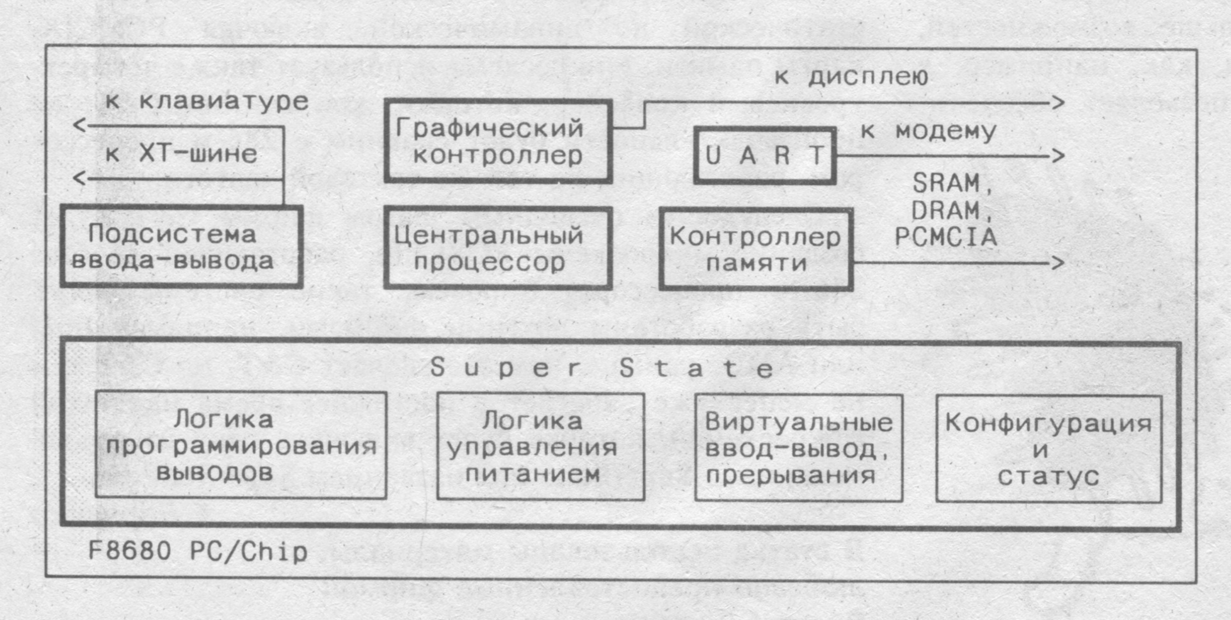


Рис.3

Важно отметить, что определение "PC в одной микросхеме" включает в себя только обычные общие функции, требующиеся для создания основной XT- или AT-совместимой системы. То есть, как правило, имеются в виду сам микропроцессор, контроллер дисплея, памяти, диска, система ввода-вывода и контроллер клавиатуры.

Другими функциями, которые, возможно, будут добавлены к PC/Chip следующего поколения, могут быть кэш, математический сопроцессор, акселераторы для специальных функций, таких как графические устройства в Microsoft Windows и контроллеры устройств, использующих интерфейс SCSI. В дальнейшем PC/Chip сможет включать функции мультимедиа и сжатия данных или распознаватель голоса.

Существует целый ряд причин, по которым производители микросхем стараются выйти на рынок именно с продукцией однокристального компьютера. И важнейшая из них — интеграция. Понятно, что проще иметь дело с одной микросхемой, чем с десятью, кроме того, это значительно облегчает работу проектировщиков компьютерных систем. Интеграция системы непосредственно влияет на ее стоимость, потребляемую мощность, размеры и вес.

Ожидается, что в целом стоимость PC в одной микросхеме будет меньше, чем общая стоимость входящих в него частей. И эта стоимость будет понижаться и в дальнейшем.

Несомненно, большое значение имеет и размер всей системы. Хотя рынок компьютеров типа ноутбук огромен, но пользователи, тем не менее, не заинтересованы в том, чтобы уменьшение размеров устройств происходило за счет сокращения их возможностей. PC/Chip значительно уменьшает размер системной платы и позволяет производителям, с одной стороны, выпускать системы, меньшие по размеру, с другой — "втиснуть" в них еще больше функций.

Меньший размер влияет также и на потребляемую мощность. Многие пользователи хотят иметь портативные переносные системы для того, чтобы использовать их во время путешествий, в пути, когда энергопитание компьютера может осуществляться только от аккумуляторов. Низкое потребление мощности PC/Chip в этом

случае может значительно продлить период работы системы.

Другое преимущество состоит в том, что производителям уже не надо разрабатывать никаких вспомогательных узлов компьютера и они могут, например, позаботиться о других особенностях своей системы.

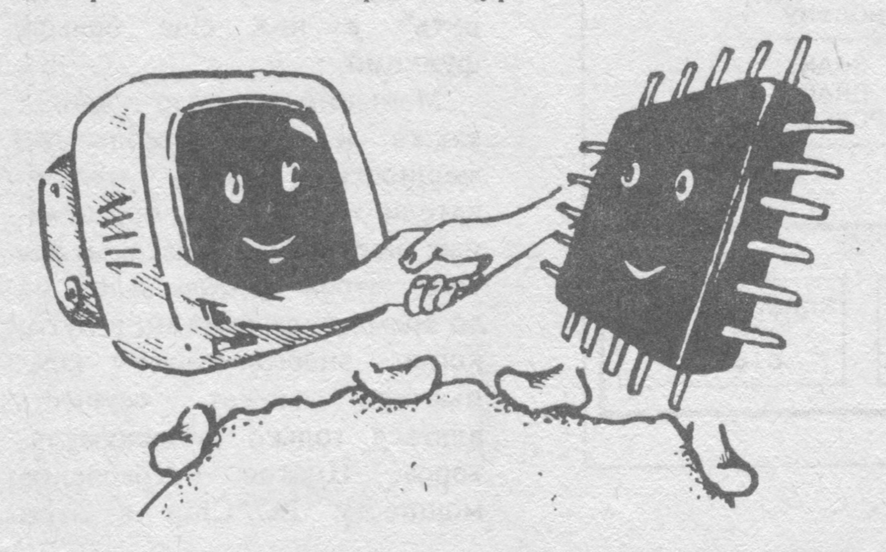
Все компании, поставившие своей целью создание PC в одной микросхеме, идут к ее осуществлению разными путями. Возможно, наиболее значительным достижением в этом плане можно считать разработки фирмы Intel. Ей уже удалось создать свое решение "неразрывно связанного" набора микросхем i386SL.

Фирма Intel объявила о своем намерении войти в рынок AT-микросхемой. Компания не указала, какими конкретно особенностями будет обладать AT-микросхема. Вероятно, она будет снабжена основным набором базовых функций и сможет представлять интересы Intel на рынке продуктов высшего класса, имея в этом наборе такие особенности, как кэш, функции управления для VGA и, возможно, даже математический сопроцессор. Такие известные фирмы, как AMD и IIT, судя по всему, также работают над созданием PC в одной микросхеме.

То, что C&T стала первой компанией, которая анонсировала PC/Chip, не вызывает особого удивления, если принять во внимание богатый опыт компании на рынке полупроводниковых микросхем. Однако архитектурные новшества, которые включает в себя эта микросхема, поистине удивительны.

Микросхема F8680 PC/Chip, по сути, состоит из микропроцессора 8086, таймера 8524, контроллера прерываний 8259, контроллера памяти с обширными возможностями, механизма эмуляции DMA, контроллера клавиатуры для XT, графического контроллера для CGA и универсального асинхронного приемопередатчика 16C450 для последовательной связи. Эта микросхема поддерживает также стандартную восьмибитную шину XT. Примерная функциональная схема PC/Chip приведена на рис. 3.

Как уже говорилось, эта микросхема имеет новый операционный режим, называемый SuperState R. В результате этого создается гибкая архитектура, которая предоставляет F8680 гораздо больше возможностей, чем только управление питанием, как, например, в наборе i386SL. Эта архитектура позволяет обеспечивать виртуальные устройства, осуществлять управление новой внешней аппаратурой и эмулировать другие процессоры. Один из первых покупателей F8680 планирует создать систему, которая будет эмулировать IBM AT, используя режим SuperState.



Графический контроллер в F8680 поддерживает электронно-лучевые мониторы, плоские LCD-экраны и является полностью совместимым с CGA, поддерживая разрешающую способность 640 на 200 или на 400 пикселов в монохромном или 320 на 200 или на 400 пикселов в четырехцветном изображении. Кроме того, фирма C&T включила алгоритм Visual Map, который позволяет контроллеру управлять LCD-экраном, достаточно точно передавая цветное изображение в 16 оттенках серого цвета.

Управление питанием является важнейшей частью разработки микросхемы F8680. Частично оно выполняется, используя режим SuperState, частично благодаря аппаратным особенностям самой микросхемы F8680. Ядро микропроцессора может работать, используя любое напряжение от 2,7 В до стандартного напряжения в 5 В, в то время как дополнительные элементы для создания системы на базе F8680 могут работать с напряжением только 5 В. В настоящее время полупроводниковая промышленность движется в сторону нового стандарта питания 3,3 В. Микросхема, работающая под напряжением 3,3 В, использует приблизительно 40% мощности, потребляемой при тех же условиях микросхемой, рассчитанной на напряжение 5 В. Особенности разработки микросхемы F8680 дают возможность полностью остановить задающий генератор системы, временно приостановив работу системы. Кроме того, тактовый сигнал может быть поделен, вследствие чего система работает медленнее, но потребляет при этом меньше мощности. Судя по техническим данным, PC/Chip работает на частоте от 0 до 14 МГц.

Одно из важнейших преимуществ модели F8680 по сравнению с 8086 или даже 286 состоит в том, что она использует 26-битное адресное пространство, позволяющее получить доступ к 64 Мбайт памяти. Микросхема может работать со всеми видами памяти — от статической до динамической, включая PCMCIA-карты памяти. Микросхема использует также четырехуровневый конвейер, который, как считает C&T, по производительности будет сравним с 286-м процессором, работающим на той же тактовой частоте.

Следующим очевидным шагом фирмы C&T будет создание микросхемы PC/Chip, работающей на базе 386-го процессора. Впрочем, такие системы могут быть разработаны другими фирмами, например Intel или AMD, раньше, чем это сделает C&T, но C&T тем не менее уже работает в настоящее время над такой микросхемой, которая будет включать одну из версий механизма SuperState под названием SuperState V.

А.Борзенко

В статье использованы материалы, любезно предоставленные фирмой Summit Systems

Приверженцы Windows неоднократно сетовали на практически полное отсутствие пакетов баз данных, работающих в этой графической среде. В предлагаемой вашему вниманию заметке мы публикуем краткое описание разработки фирмы Software Products International, СУБД WindowsBase, работающей в среде Microsoft Windows.

WindowsBase: баз данных прорубив окно

Фирма Software Products International приступила к продажам на российском рынке системы WindowsBase — мощной реляционной СУБД, имеющей полную реализацию языка SQL и разнообразные средства разработки приложений в среде Windows.

Система WindowsBase поддерживает реляционные базы данных, основное преимущество которых состоит в простоте поиска и манипулирования данными и возможности по мере необходимости объединять различные данные, не прибегая к процедурным средствам языков программирования. Помимо непосредственного доступа к базам данных система обеспечивает быструю разработку приложений с экранными формами, отчетами и меню.

База данных WindowsBase состоит из таблиц с определенным числом столбцов и неограниченным числом строк, данные которых могут динамически связываться. Таблицы бывают исходные, генерируемые по запросу, временные и отражающие виртуальное представление пользователя.

В системе реализованы шлюзы, обеспечивающие создание и обработку таблиц в форматах xBase, Btrieve и Open Access. Имеются развитые средства импорта и экспорта данных.

WindowsBase поддерживает строки длиной до 254 символов фиксированной и переменной длины, битовые строки, 4 типа числовых данных (короткое целое, целое, десятичные числа и числа с плавающей точкой), три типа данных для хранения дат и времени, а также примечания.

Один или несколько столбцов таблицы могут объявляться ключами индекса. Индекс может быть объявлен как уникальный (система при этом отклоняет ввод уже существующих его значений).

Представлениями называют не существующие физически, но применяемые пользователем таблицы. Определение представления включает перечень столбцов, которые могут быть ссылками на строки других таблиц, а также выражениями или функциями, в которых упоминаются имена различных столбцов.

В системе имеется центральный словарь данных, также реализованный в виде совокупности таблиц (syscolauth, syscolumns, sysdatabases, sysdepend, sysindexes, sysrules, systabauth, systables, systypes и sysusers). По мере создания и модификации объектов базы данных их описания заносятся в словарь. Таблицы словаря пользователь может просматривать (но не корректировать).

WindowsBase имеет несколько функциональных модулей (процессоров), предназначенных для выполнения различных функций разработки приложения.

Процессор форм. Обеспечивает доступ к базе данных через экранные формы. В начале работы открывается файл описания формы, и после появления формы на экране манипулирование данными осуществляется с помощью включенных в форму управляющих элементов оконного интерфейса (кнопок, линеек просмотра и т.д.). Процессор форм позволяет реализовывать достаточно сложные приложения, работая в которых пользователь может инициировать процедуры обработки данных простым нажатием кнопки.

Процессор таблиц. В режиме доступа к таблицам работа ведется в блоках диалога, позволяющих посредством ответа на подсказки описывать запросы. Результаты обработки запроса также выводятся на экран в табличной форме.

Процессор запросов. В этом режиме пользователь сам вводит операторы SQL (или загружает готовые текстовые файлы с запросами), а результат выводится на экран и при необходимости может быть записан в текстовый файл. Работа здесь требует знания языка SQL.

Операторы SQL реализуют функции:
- описания и изменения структуры таблиц, индексов и представлений (CREATE TABLE, CREATE INDEX, DROP и ALTER TABLE);

3. Зак.3010

- манипулирования данными (SELECT, UPDATE, INSERT и DELETE);
- управления доступом и контроля полномочий пользователей (GRANT и REVOKE).

Генератор отчетов. В этом режиме пользователь, загружая ранее созданные файлы макетов отчетов, может распечатывать выходные документы, содержащие актуальные значения полей базы данных.

Конструктор таблиц позволяет просматривать описания таблиц, создавать и изменять их структуру, а также строить индексы.

Конструктор форм предоставляет средства ускоренной разработки приложений, не требующие программистских навыков.

Конструктор отчетов. Здесь пользователь создает макеты документов, выводить которые можно в режиме получения отчетов. Пользовательский интерфейс при конструировании форм и отчетов практически одинаков.

Сервис. WindowsBase располагает сервисными функциями как для администратора, так и для пользователей прикладных систем. Имеются средства управления доступом, сопровождения базы данных, экспорта и импорта данных и др.

Компоновщик приложений. Этот режим позволяет непрограммисту быстро создавать меню в стиле Windows, с помощью которых нетрудно связать в единую прикладную систему имеющиеся формы, отчеты и запросы.

Прикладное меню. В этом режиме пользователь загружает созданные им файлы меню и получает возможность работать в собственных прикладных системах.

Меню настройки доступно в любом режиме работы и позволяет оперативно изменять параметры конфигурации системы (пути доступа, стандартные форматы и т.п.).

Механизм обработки транзакций позволяет организовать дополнительный контроль за внесением изменений в таблицы. Транзакция — это некоторая последовательность операторов SQL, модифицирующих данные таблиц. Основной отличительной особенностью транзакции является то, что она должна либо полностью выполняться, либо не выполняться вовсе.

Начало транзакции отмечается оператором begin transaction. При выполнении транзакции изменения сначала заносятся в специальный файл журнала, а запись их в базу данных выполняется оператором commit transaction, завершающим транзакцию. Им пользуются после того как установлено, что все логически связанные изменения выполнены успешно и нужно сделать их постоянными. Оператор rollback при необходимости возвращает таблицы базы данных в первоначальное состояние.

WindowsBase располагает средствами разграничения и управления доступом пользователей к базе данных. В словаре данных хранится информация обо всех таблицах, столбцах, индексах, пользователях и их взаимосвязях. Администратор базы данных имеет возможность динамически определять полномочия пользователей по выполнению различных операций со всей базы данных (открытие, изменение таблиц), отдельными таблицами (чтение, запись, корректировка, удаление строк, изменение структуры) и даже столбцами таблиц.

Интерфейс системы соответствует среде Windows. Меню режимов работы позволяет быстро переходить от одного вида деятельности к другому, выбирая соответствующую пиктограмму. Система контекстной подсказки активизируется при выборе пункта меню "Справка" или нажатии клавиши F1. Текст справок находится в плоском файле, который пользователь может корректировать и дополнять.

WindowsBase — мощная система и для ее производительной работы требуется не менее 4 Мбайт оперативной памяти и достаточно быстрый винчестер.

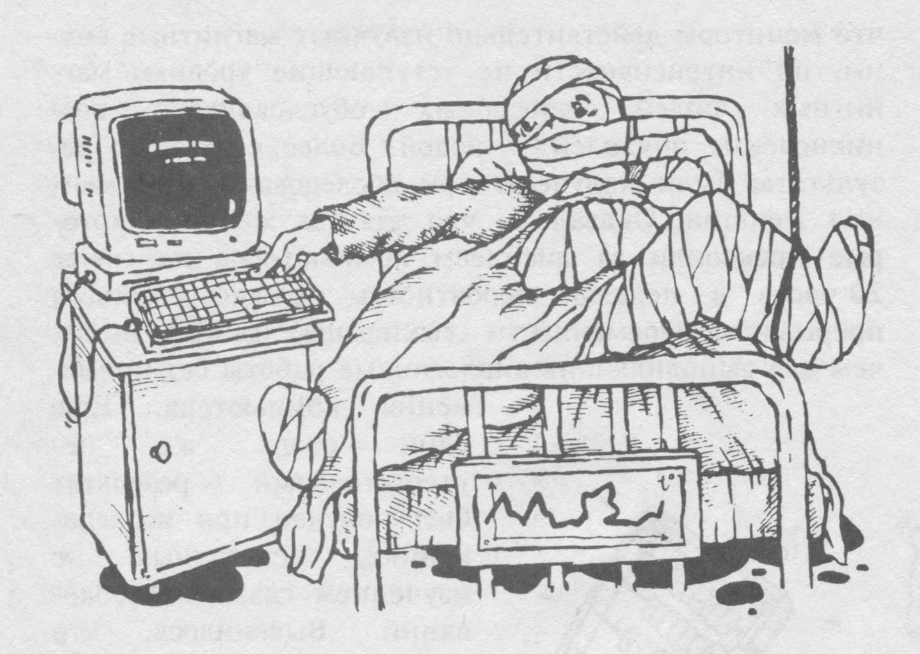
Поставляется система на двух дискетах либо 5,25" либо 3,5".

Документация включает три книги и информационные материалы по текущей версии.

Стоит однопользовательская (оригинальная) версия 350 долл. США или 48 тыс. руб.

В октябре на выставке SofTool'92 будет представлена русская версия продукта.

А.Александров



Работа с персональными компьютерами доставляет немалое удовольствие как тем, кто только начинает их осваивать, так и тем, кто уже освоил их настолько, что не в силах оторваться от клавиатуры и дисплея. Однако интенсивная работа с персональными компьютерами может причинить и немало неприятностей. Связаны они обычно с незнанием мер безопасности или пренебрежением ими.

Губит людей не пиво...

При использовании человеком даже самой передовой технологии у него могут возникнуть соответствующие профессиональные заболевания, если, работая, он будет пренебрегать даже элементарными правилами техники безопасности. В данной статье рассматриваются основные симптомы и причины тех заболеваний, которые могут возникнуть у чрезмерно увлеченных и беспечных пользователей персональных компьютеров.

Типичными ощущениями, которые испытывают к концу рабочего дня фанатики персональных компьютеров, являются: головная боль, резь в глазах, тянущие боли в мышцах шеи, рук и спины, зуд кожи на лице и т.д. Испытываемые день за днем, они могут привести к мигреням, частичной потере зрения, сколиозу, тремору, кожным воспалениям и другим нежелательным явлениям. Все это не случайно. В США зарегистрированы десятки случаев возникновения болезней, причины которых связывают с работой на компьютере. Уже выплачены сотни тысяч долларов по искам о возмещении убытков из-за потери трудоспособности при работе за терминалами компьютеров, подсчитаны и немалые убытки, вызванные потерями рабочего времени. По данным Национальной академии наук США, а также по результатам многих исследований, проведенных учеными Австралии, Германии и ряда международных центров, выявлена определенная связь между работой на компьютерах и такими недомоганиями, как астенопия (быстрая утомляемость глаз), боли в спине и шее, запястный синдром (болезненное поражение срединного нерва запястья), тендениты (воспалительные процессы в тканях сухожилий), стенокардия и различные стрессовые состояния, сыпь на коже лица, хронические головные боли, головокружение, повышенная возбудимость и депрессивные состояния, снижение концентрации внимания, нарушение сна и, к сожалению, немало других, которые не только ведут к снижению трудоспособности, но и подрывают здоровье людей.

Вероятнее всего, человеку уже никогда не удастся полностью избежать пагубного влияния передовых технологий, но, как и во многих других случаях, сами пользователи персональных компьютеров, по крайней мере, могут свести их к минимуму. Большинство проблем решаются сами собой при правильной организации рабочего места, соблюдении правил техники безопасности и разумном распределении рабочего времени.

Основным источником эргономических проблем, связанных с охраной здоровья людей, использующих в своей работе автоматизированные информационные системы на основе персональных компьютеров, являются дисплеи (мониторы), особенно дисплеи с электронно-лучевыми трубками. Они представляют собой источники наиболее вредных излучений, неблагоприятно влияющих на здоровье операторов.

Как отметил канадский биофизик К.Мара, “нет ни одного вопроса столь важного, волнующего и сложного, как вопрос электромагнитного излучения дисплея“. История исследования этого вопроса достаточно длительная и непростая, но полученные результаты

носят пока еще преимущественно статистический характер и не имеют адекватного объяснения. Частотный состав (спектр) излучения монитора характеризуется наличием рентгеновских, ультрафиолетовых, инфракрасных и других электромагнитных колебаний. Опасность рентгеновского и части других излучений большинством ученых признается пренебрежимо малой, поскольку их уровень достаточно невелик и в основном поглощается покрытием экрана. Наиболее тяжелая ситуация связана, по-видимому, с полями излучений очень низких частот, которые, как выяснилось, способны вызывать биологические эффекты при воздействии на живые организмы. Было обнаружено, что электромагнитные поля с частотой порядка 60 Гц могут инициировать изменения в клетках животных (вплоть до нарушения синтеза ДНК). Особенно поразительным для исследователей оказался тот факт, что, в отличие, например, от рентгеновского излучения, электромагнитные волны обладают необычным свойством: опасность их воздействия при снижении интенсивности излучения не уменьшается, мало того, некоторые поля действуют на клетки тела только при малых интенсивностях или на конкретных частотах. Согласно одному из объяснений, сформулированных американскими учеными, переменное электромагнитное поле, совершающее колебания с частотой порядка 60 Гц, вовлекает в аналогичные колебания молекулы любого типа, независимо от того, находятся они в мозге человека или в его теле. Результатом этого является изменение активности ферментов и клеточного иммунитета, причем сходные процессы наблюдаются в организмах и при возникновении опухолей. “Не нужно быть медиком, — замечает американский врач П.Броудер, — для того, чтобы понять, что подобные электромагнитные явления, не имеющие аналогов в истории эволюции человека, вполне могут оказывать вредное влияние на организм“. Специальные измерения показали, что мониторы действительно излучают магнитные волны, по интенсивности не уступающие уровням магнитных полей, способных обусловливать возникновение опухолей у людей. Более серьезные результаты были получены при обследовании беременных женщин. Оказалось, что для тех женщин, которые проводили за дисплеем компьютеров не менее 20 часов в неделю, вероятность преждевременного прерывания беременности (выкидыша) на 80% выше, чем для выполняющих аналогичные работы без применения компьютера. Еще один столь же неутешительный результат был получен при исследованиях, связанных с изучением глазных заболеваний. Выяснилось, что служащие, работающие за дисплеем компьютера по 7 и более часов в день, страдают воспалениями и другими заболеваниями глаз на 70% чаще тех, кто проводит за дисплеем меньше времени.



Технические характеристики дисплеев (разрешающая способность, яркость, контрастность, частота обновления или мелькания) в том случае, если на них не обращают внимания при выборе устройства или неправильно устанавливают, могут крайне отрицательно сказаться на зрении.

Что касается других функциональных нарушений, то те из них, которые связаны со скелетом человека, обусловлены длительными статическими нагрузками, вызванными плохой организацией рабочего места пользователя: неудобной или неподходящей по размерам мебелью, неудобным взаимным расположением компонентов системы персонального компьютера или отсутствием достаточного для свободных движений и смены позы места. Неудачная организация клавиатуры либо неудобная конструкция мыши способны вызвать “накапливание“ заболеваний сухожилий, мышц и нервных окончаний. Такие примеры, к сожалению, уже известны (правда, также за рубежом). Кроме того, возникновение болезней спины, шеи и рук специалисты объясняют тем, что при работе с клавиатурой

компьютера пользователи с высокой скоростью повторяют одни и те же движения (типа быстрых нажатий клавиш, перемещения мыши, наклонов и поворотов головы и т.п.). Каждое нажатие на клавишу, естественно, сопряжено с множественным сокращением мышц, перемещением сухожилий вдоль костей и соприкосновениями их с внутренними тканями. В итоге из-за чрезмерной напряженности работы или увлеченности ею могут развиваться болезненные и воспалительные процессы.

Кожные заболевания (лица) связаны в основном с тем, что наэлектризованный экран дисплея притягивает частицы взвешенной в воздухе пыли, так что вблизи него "качество" воздуха ухудшается и оператор вынужден работать в более запыленной атмосфере. Кстати, таким же воздухом он и дышит.

На основе сказанного, не вдаваясь в дальнейшие анатомические и медицинские подробности, можно сформулировать определенные рекомендации для пользователей персональных компьютеров с точки зрения охраны их труда. В принципе, основной подход к решению проблем такого рода сводится к установлению строгого контроля за соответствием аппаратных и программных средств, а также условий их эксплуатации эргономическим требованиям. Это служит основой для выработки общих рекомендаций. Приведем некоторые из них:

- соблюдение ограничений по медицинским показаниям;
- внимательное отношение к характеристикам дисплея;
- правильная организация рабочего места оператора;
- правильная организация рабочего времени оператора.

Рассматривая подробно каждый их них, можно сформулировать рекомендации частного характера:

- необходимо соблюдать ограничения на работу с персональными компьютерами для служащих, страдающих заболеваниями опорно-двигательного аппарата, глаз (или нарушениями зрения), кожи, а также для беременных женщин (во всех случаях лучше получить консультацию у врача);
- предпочтительнее использовать дисплеи с высокой разрешающей способностью (разрешением) и удоб-

ным размером экрана (лучше не применять CGA-мониторы и малоразмерные, менее 14“ по диагонали, экраны);
- лучше выбирать видеоадаптеры с высоким разрешением и, по возможности (если есть на рынке и цена приемлемая), частотой обновления экранного изображения не менее 70-72 Гц;
- обязательно ставить на дисплеи экранные, в частности, поляризационные, фильтры, в несколько раз снижающие утомляемость глаз;
- если позволяют условия, сидеть не ближе 70 см (примерно на расстоянии вытянутой руки) от дисплея;
- экран дисплея не должен быть ориентирован в сторону источников света (окон, настольных ламп и т.п.);
- при размещении рабочего места рядом с окном угол между экраном дисплея и плоскостью окна должен составлять не менее 90 градусов (для исключения бликов), прилегающую часть окна желательно зашторить;
- не следует располагать дисплей непосредственно под источником освещения или вплотную с ним;
- желательно, чтобы освещенность рабочего места оператора не превышала 2/3 нормальной освещенности помещения;
- стена позади дисплея должна быть освещена примерно так же, как его экран;
- при размещении в одной комнате нескольких персональных компьютеров расстояние от рабочего места каждого оператора до задних и боковых стенок соседних персональных компьютеров должно составлять не менее 1,2 м (именно через эти стенки происходит наиболее сильное излучение от блоков развертки изображения);
- рабочее место должно быть оборудовано так, чтобы исключать неудобные позы и длительные статические напряжения тела;
- поскольку найти такое идеальное положение для тела, в котором можно было бы пребывать в течение всего дня, вряд ли возможно, для большинства людей комфортабельным может быть рабочее место, которое можно приспособить, как минимум, для двух позиций (при этом положение оборудования должно соответствовать выполняемой работе и привычкам пользователя);
- общее время работы с дисплеем не должно превышать 50% всего рабочего времени оператора;
- не следует превышать темп работы порядка 10 тысяч нажатий клавиш в час (примерно 1500 слов);
- при обычной работе с компьютером необходимо делать 15-минутные перерывы через каждые два часа, а при интенсивной работе — через каждый час;
- наконец, при вводе данных с клавиатуры рекомендуется не зажимать телефонную трубку между плечом и ухом, а также бросить курить (что вредно как для пользователя, так и для компьютера).

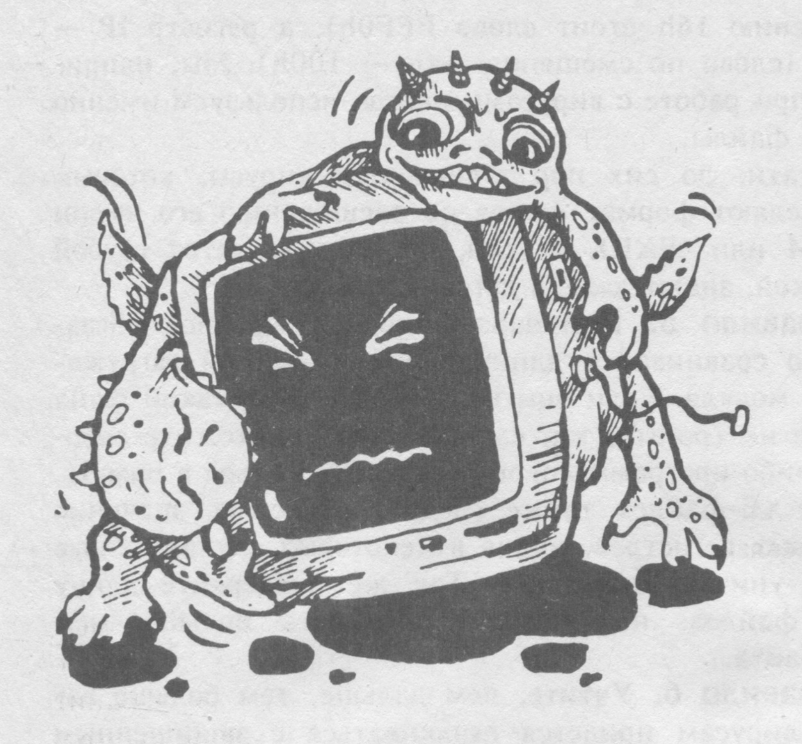
В нашей стране различными ведомствами (в основном, в сфере образования) когда-то разрабатывались временные санитарные нормы по организации работы на компьютерной технике. В этих нормативных документах приведены правила оснащения специализированных помещений, рассматриваются вопросы размещения персональных компьютеров, оформления рабочего интерьера и пр. При необходимости всегда можно обратиться к этим документам (если удастся их отыскать).

Помимо безопасности пользователя, необходимо сказать несколько слов и о безопасности компьютера. Прежде всего, нужна его защита по сети питания, а при наличии межкомпьютерных связей — и по каналам связи. Для этого применяются так называемые бесперебойные источники питания, поддерживающие напряжение питания в течение некоторого времени при аварийных ситуациях в электрической сети (этого времени обычно бывает достаточно, чтобы сохранить информацию и выключить персональный компьютер), и фильтры-стабилизаторы, защищающие от выбросов и скачков напряжения. При работе в сети можно использовать специальные платы связи, снабженные аналогичными фильтрами, защищающими “нежную“ электронику от наводок на коммуникационные линии.

Ни в коем случае нельзя загораживать заднюю стенку системного блока или ставить персональный компьютер вплотную к стене — это приводит к “тяжелому“ режиму охлаждения системного блока и его перегреву. То же самое относится к дисплею — нельзя класть на него бумаги, книги и вообще все, что может закрыть его вентиляционные отверстия. Не рекомендуется курить в помещении, где установлены персональные компьютеры, поскольку мельчайшие твердые и смолистые частицы дыма могут повредить тонкие электронные компоненты, в первую очередь, головки и механизм накопителей на магнитных дисках. Наконец, такая мелочь, о которой принято забывать каждый день: пыль и электроника плохо совместимы друг с другом, поэтому нужно почаще протирать основные устройства системы персонального компьютера, закрывать их полиэтиленовыми чехлами по окончании работы и вообще следить за толщиной слоя пыли в рабочем помещении. Наконец, не рекомендуется часто переставлять компоненты системы персональных компьютеров в помещении, в частности, таскать с места на место системный блок (если уж возникает необходимость в этом, то надо, как минимум, “запарковать“ головки дискового накопителя, аккуратно отсоединить все кабели и по отдельности перенести процессорный блок, дисплей, принтер и пр.).

Соблюдение перечисленных в этой статье элементарных правил позволит сохранить как здоровье пользователям персональной компьютерной техники, так и ей самой.

Т.Алимова, Л.Лямин



Несколько советов авторам вирусов

Успокойтесь! Не надо готовить ругательства или, наоборот, потирать руки. Мы не хотим делиться своими идеями с авторами компьютерных вирусов. Все значительно проще — через наши руки прошло несколько сотен образцов компьютерных животных, и слишком часто в них встречались одни и те же ошибки. С одной стороны, это хорошо — такие вирусы часто оказываются "маложивущими", но с другой стороны, малозаметная ошибка может привести к несовместимости вируса и используемого на компьютере программного обеспечения. В результате вирус "вешает" систему, компьютер отдыхает, а пользователи мечутся в панике с криками: "Пусть хоть 100 вирусов, лишь бы компьютер работал!!!" (завтра сдавать заказ, не запускается самая любимая игрушка, компилятор виснет при выходе в DOS и т.п.). И все это происходит при заражении довольно безобидным вирусом.

По причине этого и возникло желание поделиться некоторой информацией о жизни вируса в компьютере, дабы облегчить жизнь и вам, и многочисленным "пользователям" ваших вирусов. Итак, правила для автора вируса.

> Умышленное уничтожение или повреждение государственного или общественного имущества —
> наказывается лишением свободы на срок до одного года, или исправительными работами на тот же срок, или штрафом, или возложением обязанности загладить причиненный вред.
>
> (Ст.98 УК РСФСР)

Правило 0. Не надо писать вирусы — на свете так много интересного! Если нечего делать, то перепишите что-нибудь с DOS на Windows, если хочется прославиться — сделайте то же самое, но потом раздайте свою программу всем желающим и нежелающим.

Правило 1. При запуске вирус должен сохранять все регистры, которые передает DOS загружаемой программе (то есть не только AX, DS и ES, но и остальные), а при переходе на программу восстанавливать их (за исключением корректировки SS:SP, если это нужно). Почему? Объясняем.

Существуют (и обязательно появятся еще) программы, которые используют для своих нужд значения регистров, передаваемых из DOS´a при загрузке. Например, довольно популярный архиватор COM- и EXE-файлов DIET (версия 1.10a) при распаковке запускаемых COM-файлов предполагает, что значение регистра BX обязательно должно быть равно 0000h. Если подобная программа будет заражена вирусом, который не восстанавливает регистры при передаче управления на тело программы, то она скорее всего "умрет" при загрузке.

При загрузке регистры принимают значения (верно для MS-DOS 3.30 — 5.0):

AX = BX — корректность имени диска в командной строке (0 или 00FF); CX, BP — не определены; DX=DS; SI=IP; DI=SP.

Правило 2. Если вы используете или корректируете системные области DOS (буферы, стек, фрагменты обработки прерываний), вы обязательно должны проверять, в какой системе находитесь. Почему? Объясняем.

Как правило, каждая следующая версия DOS своей внутренней организацией не похожа на предыдущую. Это касается и буферов, и стека, и многого другого. Поэтому, если вы полностью дизассемблировали и разобрали по косточкам версию MS-DOS 3.30, это во-

все не значит, что вы постигли все тайны фирмы Microsoft и вам можно ставить памятник. Нет! Вам следует перекурить и приниматься за версию 4.0x, потом 5.00, а в минуты отдыха и по ночам потрошить DR-DOS.

Правило 3. При заражении программ сначала записывайте вирус, а затем модифицируйте начало файла (в случае, если вирус пишется в начало или середину файла — сначала сохраните стираемую часть, а затем записывайте вирус). Почему? Объясняем.

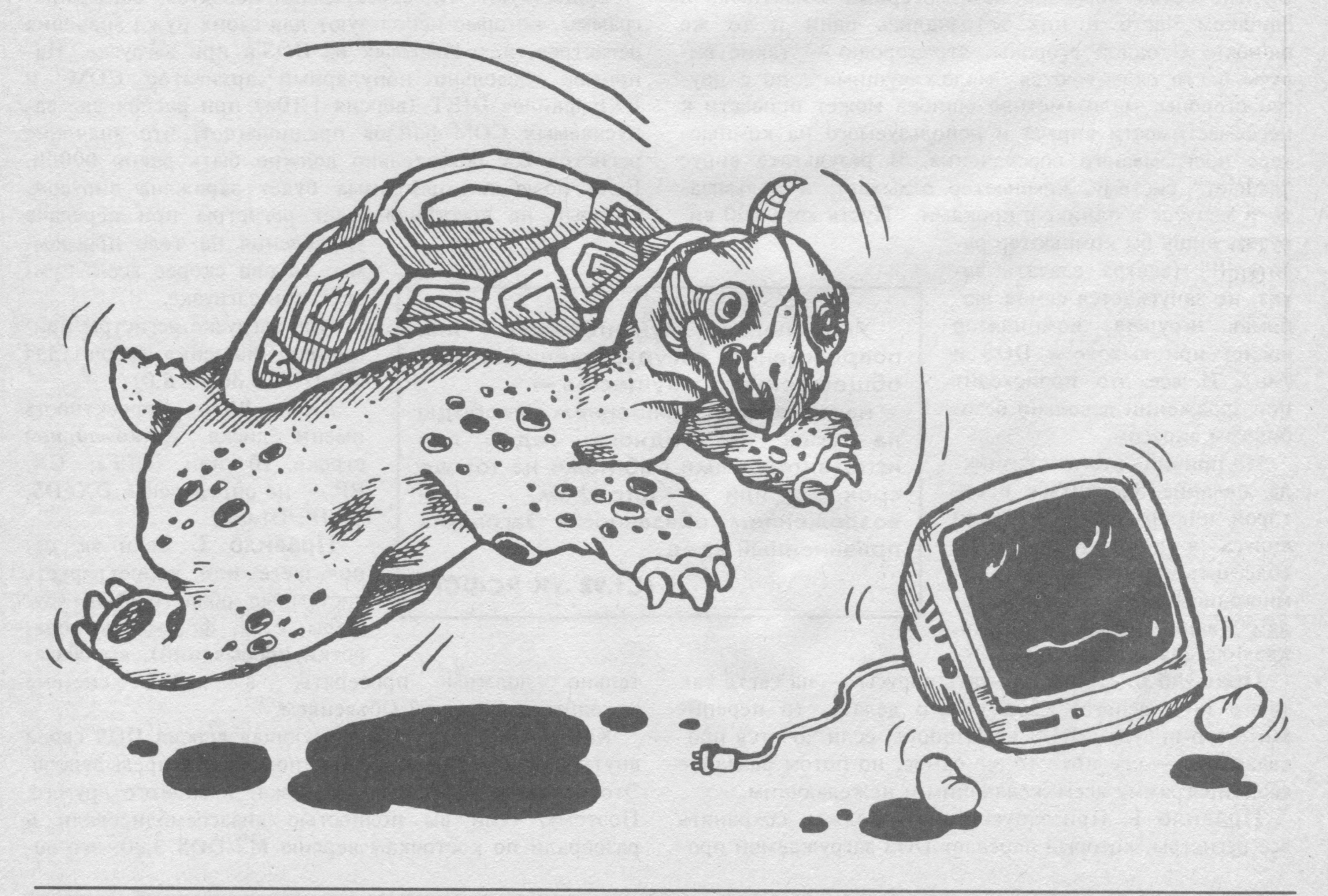
При записи в файл может произойти ошибка (кстати, не забывайте проверять флаг CF при операциях с дисками), и если ваш вирус модифицировал начало файла, а сам не записался, то файл безнадежно испортится и при загрузке скорее всего зависнет.

Правило 4. Если вы заражаете COM- и EXE-файлы, сохраняйте первые два байта заражаемой программы и проверки "COM или EXE" проводите только по ним (напомню — в начале EXE-файлов должна быть сигнатура — 4D5Ah или 5A4Dh). Говорим об этом только потому, что участились случаи появления вирусов, которые проверяют "COM или EXE" по способу: равен регистр CS регистру DS (ES) или нет (некоторые даже умудряются проверять по регистру IP, — равен он 100h или нет). Это в корне неверно, так как при запуске EXE-файла регистр CS может быть равен регистру DS (если в EXE-заголовке по смещению 16h стоит слово FFF0h), а регистр IP — 100h (слово по смещению 14h — 100h). Мы, например, при работе с вирусами иногда используем именно такие файлы.

Кстати, до сих пор попадаются вирусы, которые определяют формат файла по расширению его имени (.COM или .EXE). О том, что это является грубой ошибкой, знают даже в детских садах!

Правило 5. При заражении EXE-файлов обязательно сравнивайте длину файла с длиной загружаемого модуля. Если они не совпадают, то такой файл лучше не трогать, так как вы либо уничтожите оверлей, либо программа перестанет загружаться в память. Для EXE-файлов также следует проверять значения стековых регистров, иначе в некоторых случаях вирус будет уничтожен стеком. Так же проверяйте длину COM-файлов: не следует их делать больше, чем 64 Кбайта.

Правило 6. Учтите, чем дальше, тем больше вашим вирусам придется сталкиваться с защищенным режимом процессоров 80x86, в котором, как известно, перехват прерывания 21h является довольно тонким вопросом. Если вы перехватываете int 21h "по-старинке", то многие новые продукты (например, Borland C++ 3.0) будут зависать, а Windows при нескольких DOS-задачах в фоновом режиме может выдать "unrecoverable application error".





Для того, чтобы избежать этого, помимо всех известных правил подмены критических прерываний (установка флагов и регистров на выходе), существуют еще два:
- непосредственно перед вызовом OLD_21 нужно запретить аппаратные прерывания (cli);
- для всех функций, которые не обрабатывает ваш резидент, OLD_21 нужно выполнять через JMP, а не через CALL.

Правило 7. Теперь насчет порчи программ и дисков. Сравните вирусы зарубежного и отечественного производства. Вам сразу бросится в глаза, что почти каждый советский вирус, в отличие от зарубежных (не считая тайваньских: "Восток — дело тонкое"), что-нибудь, да испортит. Даже большая группа болгарских вирусов не идет ни в какое сравнение с нашими вирусами по части пакостей. Что это — особенности русской (точнее — советской) души или наследие сталинизма?

Короче, если вы пишете вирус, старайтесь воздерживаться от вставки пакостей — это может вам же выйти боком. Несмотря на отсутствие в нашем уголовном законодательстве статьи о написании вирусов, там есть другая статья — "причинение материального ущерба". Вот по этой статье вас и посадят, тем более, что такой случай уже был (по сообщению Н.Н.Безрукова). Ну, а если и не посадят, то пострадавшие могут найти и, извините, морду набить. И такие случаи тоже бывали.

Ну а уж если вы все-таки вставили какую-то гнусность, то запускайте свой вирус на своем компьютере и наслаждайтесь.

В.Богданов,
А.Де-Мондерик,
Е.Касперский

Каталог продуктов фирмы NOVELL

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СЛУЖБ ПЕЧАТИ И ФАЙЛОВ

NetWare for VMS

NetWare for VMS v2.1 — это программный продукт, позволяющий системам VAX/VMS фирмы DEC функционировать в качестве серверов NetWare, обеспечивая "прозрачную" интеграцию локальных сетей, объединяющих персональные компьютеры, и сетей мини-ЭВМ, объединяющих машины VAX, работающие под операционной системой VMS. Пользователи DOS и OS/2 могут использовать данные, прикладные программы и службы печати совместно с пользователями терминалов VAX. "Прозрачный" доступ пользователей персональных компьютеров к данным и принтерам VMS обеспечивается службами печати и файлов. Служба эмуляции терминала Terminal Emulation Service (TES) позволяет пользователям осуществлять доступ из среды DOS к прикладным программам VMS.

NetWare for VMS — это инструмент обеспечения связи на уровне всего предприятия. Он позволяет супервизорам сети объединять персонал, рабочие группы, компьютеры, использующиеся в отделах и на предприятии в целом, в рамках общей сетевой операционной системы. Такая интеграция позволяет пользователям персональных машин осуществлять доступ к информации, хранящейся в машинах VAX, принадлежащих отделам и предприятию, используя "родной" интерфейс. Пользователи ПЭВМ могут хранить информацию в среде VAX, обладающей быстрой, большой по объему и надежной дисковой памятью, которая к тому же резервируется. Эта информация хранится в виде стандартных файлов RMS, причем осуществляется автоматическое динамическое преобразование атрибутов файлов и их наименований, которые различны в операционных системах DOS, OS/2 и VMS. Между IBM PC и VAX может осуществляться обмен файлами со скоростью, обеспечиваемой используемой ЛВС. NetWare for VMS реализует файловые службы NetWare на VAX. Этот продукт работает как один процесс VAX, обеспечивая одновременный доступ 254 пользователей DOS и OS/2. Многопользовательская, многопоточная архитектура позволяет одновременно обрабатывать большое число запросов.

NetWare for VMS не создает проблем для управления системой, осуществляемого администратором VAX. Поскольку данные хранятся на VAX в формате RMS, NetWare for VMS подчиняется всем требованиям к защите информации, предъявляемым в системах VAX, в ней обеспечиваются собственные средства защиты информации NetWare Level I и поддерживаются средства безопасности NetWare v2.1, такие как проверка защиты каждые полчаса, выявление несанкционированного подключения и блокировка, проверка бюджетов пользователей, шифрование паролей, а также проверка права подключения в определенные часы и к определенным узлам. Пользователи ПЭВМ получают только те права доступа к файлам, которые присвоит им администратор системы VAX.

Материал подготовлен к публикации совместно СП "ИнтерПроКом" (Тел.: 129-80-09, 128-80-33) и агентством КомпьютерПресс. Начало в №№3-8´92.

Характеристики

- Совместимость. NetWare for VMS совместим с серверами сети IBM PC, работающими в операционной системе NetWare версии 2.0 и выше, включая NetWare 386.
- Прозрачное совместное использование файлов. Пользователи ПЭВМ и терминалов VAX/VMS могут совместно применять файлы и данные. Автоматическое преобразование файлов обеспечивает представление файлов в виде DOS-файлов для пользователей ПЭВМ и в виде VMS-файлов для пользователей VAX.
- Прозрачные службы печати. NetWare for VMS посылает запросы на печать с ПЭВМ в очереди печати VAX/VMS. Пользователи NetWare for VMS могут применять все средства печати Advanced NetWare v2.1, включая команды CAPTURE и NPRINT.
- Виртуальная дисковая память. Файловая система VAX представляется в виде виртуального расширения дисковой памяти ПЭВМ. В среде NetWare for VMS каждый файл хранится на ЭВМ VAX в стандартном формате RMS. Файлы, созданные в операционных системах DOS или OS/2, могут использоваться пользователями и процессами VMS.

Поддерживаются потоковые файлы и файлы в формате с фиксированной длиной записи.

- Служба эмуляции терминала Terminal Emulation Service (TES). TES — это служба связи, обеспечивающая стандартный интерфейс (по прерыванию 14), позволяющий программам эмуляции терминала различных производителей выполняться на рабочих станциях. TES поддерживает многие популярные пакеты эмуляции терминала. TES использует всего 33 Кбайта памяти рабочей станции и может быть выгружена, если она не используется. Она также может вызываться из командной строки, что позволяет начинающим пользователям легко устанавливать ее при помощи командных файлов.
- Низкие издержки VAX. Поскольку NetWare for VMS выполняется как один процесс VAX, системные издержки невелики. Вместе с NetWare for VMS на VAX могут выполняться другие прикладные программы.
- Независимость от аппаратных средств. NetWare for VMS использует стандартные котроллеры DEC VAX Ethernet, что позволяет избежать дополнительных расходов и эксплуатационных проблем, возникающих при использовании контроллеров Ethernet других поставщиков.
- Независимость от топологии. Используя мосты NetWare, NetWare for VMS поддерживает ПЭВМ, соединенные с сетями различных топологий, включая сети, отличные от Ethernet, — такие как AT&T StarLAN, IBM PC Network, IBM Token-Ring и Arcnet. Пользователи этих сетей могут осуществлять доступ к VAX точно так же, как к любому серверу NetWare.
- Поддержка платы DEPCA. NetWare for VMS включает драйвер для поддержки контроллера DEPCA PC Ethernet фирмы DEC, позволяющий пользователям, реализующим конфигурацию связи персональных компьютеров и VAX типа PCSA-DEC, переключиться на использование NetWare for VMS, не покупая новых адаптеров для рабочих станций.
- IPX через DECnet. NetWare for VMS включает процесс, обеспечивающий транспортирование через DECnet, что дает пользователям прозрачный доступ к удаленным файловым серверам VAX или NetWare, позволяя использовать широкие возможности региональных вычислительных сетей, имеющиеся в DECnet. Пакеты NetWare SPX/IPX упаковываются в пакеты DECnet и пересылаются любому файловому серверу NetWare for VMS, работающему в сети DECnet. После приема пакет DECnet распаковывается и выделенный из него пакет SPX/IPX пересылается дальше по сети к нужному серверу NetWare. Таким образом создается виртуальная сеть IPX в сети DECnet.
- Защита информации. Системы защиты NetWare и VMS работают сообща, обеспечивая гарантию от несанкционированного доступа.
- Утилиты NetWare. NetWare for VMS включает стандартные утилиты NetWare, такие как SYSCON, FCONSOLE, PCONSOLE, FILER и SESSION.

Реализация

NetWare for VMS предназначен для использования на ЭВМ VAX фирмы DEC тогда, когда необходима прозрачная поддержка пользователей персональных компьютеров. В некоторых случаях администраторам существующих сетей NetWare требуется объединить среды IBM PC и VAX. В других случаях администраторы ищут возможность объединения вновь создаваемых сетей IBM PC и VAX. В обоих случаях NetWare for VMS решает проблему.

Перед покупкой NetWare for VMS администратор сети должен определить, на какой системе VAX будет установлен данный продукт. Следует иметь в виду, что производительность NetWare for VMS зависит от мощности VAX, числа поддерживаемых клиентов NetWare и VMS, а также от того, сколько и какие прикладные программы выполняются на VAX. Хотя этот продукт рассчитан на одновременную поддержку 254 пользователей, такое число будет неприемлемым для младших поколений VAX.

NetWare for VMS поставляется на девятидорожечной магнитной ленте или кассете TK-50. Он устанавливается на VAX при помощи стандартной утилиты VMSINSTAL. Novell рекомендует проводить инсталляцию при участии дипломированного торгового агента, распространяющего продукты для VMS. Наряду с оказанием помощи в планировании и установке, агент может также предоставить техническую поддержку и неформальное обучение. Novell также обеспечивает официальное обучение работе с продуктами NetWare для VMS.

Необходимое аппаратное обеспечение

NetWare for VMS работает на любой поддерживаемой в настоящее время системе VAX или MicroVAX и требует минимум 2 Мбайта памяти (рекомендуется 4 Мбайта). Он также требует наличия на диске 10 000 блоков для работы, однако при инсталляции временно требуется 20 000 свободных блоков.

Для связи с сетью на ЭВМ VAX должен быть установлен контроллер DEC Ethernet и драйвер контроллера Ethernet, поставляемый фирмой DEC. NetWare for VMS поддерживает стандартные контроллеры DEC Ethernet, включая DEUNA, DELUA, DESQA, DELQA, DEQNA, DESVA, DEBNT, DEBNI и DEBNA.

В качестве рабочих станций сети могут использоваться IBM PC, XT, AT и совместимые ПЭВМ, а также все модели семейства IBM PS/2. Покупатели могут получить список совместимых рабочих станций у торгового агента Novell. Для каждой рабочей станции требуется сетевой адаптер, поддерживаемый драйвером NetWare v2.1. Драйвер для контроллера DEPCA Ethernet фирмы DEC входит в комплект поставки NetWare for VMS. NetWare Workstation Shell for DOS требует 55 Кбайт, а TES — 33 Кбайт оперативной памяти, но последняя может быть выгружена.

Прикладные программы DOS и другие прикладные программы требуют дополнительной памяти.

Необходимое программное обеспечение

NetWare for VMS включает все программное обеспечение, необходимое для инсталляции и работы на файловом сервере. Необходимо использование ОС VMS версии 5.x. Для поддержки рабочих станций OS/2 требуется использование NetWare Requester for OS/2. Для поддержки рабочих станций Windows/386 — NetWare Shell for Windows/386. (Только для служб печати и файлов. Для TES — не требуется.)

Модернизация

Версия 2.1 является третьей версией NetWare for VMS. Заказчики могут защитить свои вложения, воспользовавшись программой NetWare UpDate, позволяющей пользователям последней версии купить контракт на модернизацию в течение одного года. Пользователи NetWare for VMS версий 2.0a и 2.01 могут перейти к версии 2.1, воспользовавшись программой NetWare UpGrade, которая предлагает модернизацию NetWare for VMS в течение 6 месяцев без дополнительных затрат. Подробности изложены в разделах этого руководства, посвященных программам модернизации NetWare UpDate и NetWare UpGrade.

Спецификации

Поддерживаемые операционные системы рабочих станций

DOS 3.x, 4.x, 5.x
OS/2 Standard Edition 1.1
OS/2 Extended Edition 1.0, 1.1 или 1.2

Поддерживаемые программы эмуляции терминала

TES поддерживает программы эмуляции терминала других производителей, использующие прерывание 14, включая:
EM220 фирмы Diversified Computer Systems, Inc.
poly-STAR фирмы Polygon, Inc.
PC-Term фирмы Crystal Point, Inc.
Reflection series фирмы Walker, Richer and Quinn
SmarTerm фирмы Persoft, Inc.
SofTermPC фирмы Softronics, Inc.
TNET-05 фирмы Grafpoint
ZSTerm фирмы KEA Systems Ltd.

Информация для заказа

NetWare for VMS продается только через дипломированных торговых агентов Novell, занимающихся распространением продуктов для VMS. Имя вашего местного агента можно узнать по телефону Novell в США.

NetWare NFS

NetWare NFS — программный продукт, обеспечивающий прозрачный доступ рабочих станций UNIX к операционной системе NetWare v3.11. NetWare NFS состоит из нескольких загружаемых модулей NetWare (NLM), которые обеспечивают сервер NetWare возможностями сервера сетевой файловой системы Network File System (NFS). Если NetWare NFS установлен, то рабочие станции, использующие службы NFS, получают возможность совместного использования файлов с другими клиентами NetWare, включая рабочие станции DOS, Macintosh и OS/2, а также возможность печати на принтерах NetWare.

NFS, разработанная и лицензируемая фирмой Sun Microsystems, стала стандартом де-факто для распределенных файловых систем в семействе UNIX. Sun продала лицензии на технологию NFS более чем 150 поставщикам, включая всех основных производителей компьютеров. Учитывая широкое распространение NFS, Novell разработала NetWare NFS для поддержки рабочих станций UNIX в привычном для них режиме.

NetWare NFS использует те же стандартные транспортные протоколы связи, что и системы UNIX. Эти протоколы, объединенные названием TCP/IP — Transmission Control Protocol and Internet Protocol, — встроены в операционную систему NetWare v3.11. NetWare NFS предлагает следующие службы в среде TCP/IP:
- NFS для совместного использования файлов
- Line Printer Daemon (LPD) для служб печати
- File Transfer Protocol Daemon (FTPD), стандартную службу передачи файлов TCP/IP
- LOCKD, стандартную службу файлов и блокировки записей UNIX.

NetWare NFS позволяет пользователям UNIX поддерживать все средства своей операционной системы и в то же время использовать оптимизированную технологию сервера NetWare, отличающуюся высокой пропускной способностью дисковой системы, устойчивостью к сбоям и усовершенствованной системой защиты информации. Другие выгоды использования NLM NetWare NFS включают увеличение производительности рабочих групп, совместное использование ресурсов и более эффективное управление сетью.

Обеспечение клиентов UNIX прозрачным доступом к NetWare v3.11 помимо совместного использования файлов дает много других преимуществ, связанных с

производительностью и надежностью. Поскольку NetWare v3.11 превращает стандартные аппаратные средства в высокопроизводительные серверы сети, она имеет существенно лучшее соотношение цена/производительность по сравнению с традиционными серверами NFS на основе UNIX.

Характеристики

- NLM NFS Server обеспечивает механизм, позволяющий клиентам UNIX рассматривать файловую систему NetWare v3.11 как расширение "родной" среды NFS. Для доступа к томам NetWare пользователи UNIX применяют стандартную в UNIX/NFS команду монтирования тома. Ключевыми элементами NFS Server являются: прозрачная поддержка опознавания пользователя, распределение атрибутов файлов и распределение имен, блокировка файлов, распределение прав и удаленного управления с сервера.
- NLM UNIX Namespace обеспечивает файловую систему NetWare v3.11 атрибутами файлов и преобразованием наименований, использующихся в UNIX, что позволяет осуществлять прозрачный доступ к файлам из среды UNIX и совместное их использование для всей совокупности имен, поддерживаемых NetWare v3.11, включая DOS, Macintosh и OS/2.
- NLM LPD позволяет клиентам UNIX-TCP/IP посылать задания печати в обычные очереди печати NetWare v3.11.
- NLM FTPD позволяет клиентам FTP с корректным доступом к NetWare осуществлять связь с сервером NetWare v3.11 и инициировать передачу файла с любого и на любой том или директорию NetWare.
- NLM Lock Manager поддерживает блокировку файлов и записей в среде NFS. Прикладные программы, выполняющие блокировку файлов и записей UNIX, полностью поддерживаются программой Lock Manager, входящей в NetWare NFS.
- NetWare NFS имеет полный набор управляющих утилит, обеспечивающих легкость конфигурирования и контроля сервера NFS. Эти инструменты управления включают:

  Конфигурирование системы и инициализацию параметров

  Статистику использования сервера NFS

  Настройку параметров программного обеспечения для повышения производительности

  Регистрацию ошибок

  Управление правами файлов и директорий NFS.

Реализация

NetWare NFS дает пользователям рабочих групп UNIX доступ к высокопроизводительному серверу NFS. Оптимизированная операционная система NetWare v3.11 отвечает требованиям пользователей UNIX к реакции системы и высокой пропускной способности.

В сегодняшней неоднородной сетевой среде существует возможность совместного использования информации и принтеров системами, представляющими основные настольные вычислительные среды — DOS, Macintosh, OS/2 и UNIX. NetWare v3.11 и NetWare NFS связывают рабочие станции UNIX с другими операционными системами, позволяя использовать средства обработки текста, электронные таблицы, общие базы данных и сложные файлы CAD/CAM. NetWare v3.11 может использоваться как единое, централизованное хранилище файлов всех клиентов, что упрощает управление системой.

Необходимое аппаратное обеспечение

NetWare NFS работает только на машинах с процессором 386 или 486. Требуется минимум 5 Мбайт ОЗУ и 3 Мбайта на системном томе жесткого диска для установки NetWare NFS. Любая рабочая станция UNIX, поддерживающая NFS, может монтировать тома директорий NetWare NFS.

Необходимое программное обеспечение

NetWare NFS требует использования ОС NetWare версии 3.11 и выше. Для обеспечения правильной работы должен выполняться TCP NLM, входящий в комплект поставки NetWare v3.11. На машинах клиентов UNIX должно использоваться "родное" системное программное обеспечение NFS.

Опции

Готовящийся к выпуску инструментальный пакет Software Developer´s Kits (SDKs) производства фирмы Novell представляет собой средства для разработки прикладных программ NetWare. Версии SDKs для нескольких сред серверов и рабочих станций доступны через программу Professional Developers´ Program. За дополнительной информацией об этой программе и SDKs обращайтесь к Developer Relations Group. Вне территории США и Канады обращайтесь по телефону (512)794-1796 или в местное отделение Novell.

Модернизация

Novell в настоящее время продает NetWare NFS версии 1.1. Все изменения будут предоставлены участникам программы NetWare UpDate. В настоящее время программа Novell UpGrade для этого продукта не поддерживается.

Спецификации

Поддерживаемые службы

Sun Microsystems, Inc. External Data Representation Protocol (XDR)
Sun Microsystems, Inc. Remote Procedure Call (RPC)
Sun Microsystems, Inc. Network File System (NFS) version 2
Sun Microsystems, Inc. Mount Protocol
Sun Microsystems, Inc. Portmapper Protocol
Berkeley UNIX Line Printer Daemon Protocol (LPD)
DARPA (RFC 959) File Transfer Protocol (FTP)

Информация для заказа

Продукт	Номер изделия
NetWare NFS	883-001360-001

NetWare FTAM

NetWare FTAM v1.1 представляет собой сервер OSI FTAM (в терминах OSI "ответчик"), позволяющий различным клиентам FTAM ("инициаторам") осуществлять доступ к файловой системе NetWare v3.11. Этот продукт, основанный на стандартных протоколах, обеспечивает взаимодействие продуктов различных поставщиков с системами NetWare. NetWare FTAM реализует все семь уровней протоколов OSI и позволяет осуществлять обмен файлами между NetWare и любыми другими системами OSI, независимо от того, какие программные и аппаратные средства они используют. Кроме того, продукт позволяет клиентам FTAM ставить задания в очередь печати NetWare.

NetWare FTAM выполнен в виде набора загружаемых модулей NetWare (NLM), которые работают в операционной системе NetWare для процессора 80386. NetWare FTAM v1.1 полностью согласован с протоколами Government OSI Protocols (GOSIP) и используется вместе с почтовыми шлюзами X.400. Он полностью удовлетворяет основному описанию и требованиям к службам, изложенным в протоколах GOSIP 1.0. Он реализует стандартные спецификации протокола OSI для уровней FTAM, ACSE, Presentation, ASN.1, Session, Transport (Class 4) и CLNP.

Характеристики

- Обеспечивает клиентам FTAM следующие службы хранения и поиска файлов и управления в зависимости от возможностей ПО клиентов FTAM:

  Передача файлов на диск сервера для хранения
  Поиск файлов на диске сервера
  Удаление файлов с диска сервера
  Чтение атрибутов файла
  Переименование файла.
- Клиенты, использующие NBS-9, могут создавать и удалять директории и просматривать их содержимое.
- Все файлы FTAM называются в соответствии с соглашением по наименованиям UNIX.
- NetWare FTAM позволяет клиентам FTAM посылать файлы в очередь печати NetWare.

Реализация

NetWare FTAM v1.1 — идеальное решение для заказчиков, чьи сети NetWare нуждаются в согласовании со стандартами OSI, или для тех, кто просто хочет поддерживать взаимодействие продуктов нескольких поставщиков. Поставляется на дискетах размером 5,25 и 3,5 дюйма и включает утилиты инсталляции и управления.

Установленный в среде NetWare v3.11, NetWare FTAM позволяет всем совместимым системам сети осуществлять доступ к файловой системе NetWare как к виртуальным файлам FTAM. Совместимые системы включают всех клиентов FTAM, использующих протоколы GOSIP, U.K. GOSIP и MAP3.0 и поддерживающих связь через протоколы Transport Class 4, CLNP, IEEE 802.2, 803.3 и 802.5.

Необходимое аппаратное обеспечение

NetWare FTAM v1.1 требует использования операционной системы NetWare v3.11 с минимальными объемами ОЗУ и диска соответственно 6 и 1 Мбайт. В зависимости от базовой конфигурации операционной системы (количества стандартных рабочих станций DOS, загрузки сервера, объема жестких дисков) может потребоваться больший объем памяти.

Сервер должен быть соединен с сетью Ethernet, поддерживающей протокол IEEE 802.2, или с сетью

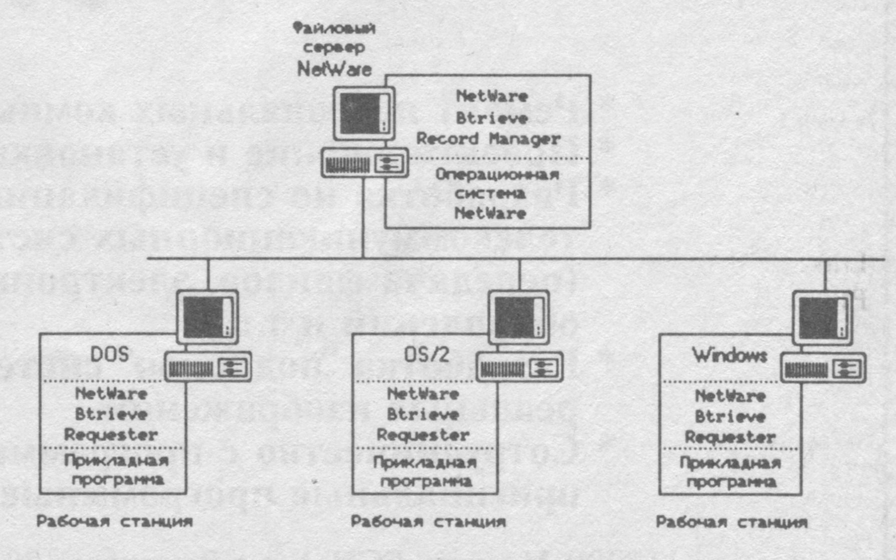


Рис. 3.22. Архитектура NetWare Btrieve

Token-Ring. При использовании одиночного коммуникационного адаптера требуется многокадровый драйвер локальной сети. Этим требованиям удовлетворяют, например, адаптеры Novell NE2000 и NE3200.

Необходимое программное обеспечение

NetWare FTAM v1.1 требует использования операционной системы NetWare v3.11, укомплектованной NLM STREAMS и CLIB. Продукт функционирует только как сервер FTAM и может быть использован совместно с любой системой FTAM, поддерживающей совместимые описания.

Спецификации NetWare FTAM v1.1

Соответствие описаниям

NIST	FTAM T1 и M1. Доступ к директориям: документ NBS-9
U.S.GOSIP	Соответствует ограниченным спецификациям FTAM Взаимодействует с полной реализацией FTAM
TOP 3.0	Соответствует ограниченным спецификациям FTAM Взаимодействует с полной реализацией FTAM
U.K. GOSIP	Взаимодействует с реализациями, соответствующими U.K. GOSIP

Соответствие стандартам

Директории:	документ NBS-9
FTAM:	ISO 8671 (файлы типов 1 и 3, ограниченное и усовершенствованное управление)
ACSE:	ISO 8650
Presentation:	ISO 8823
ASN.1:	ISO 8825
Session:	ISO 8327
Transport:	ISO 8072 (COTP), ISO 8602 (CLTP)
Network:	ISO 8473 (CLNP), ISO 9542 (ES/IS)
Link:	ISO 802.2
Physical:	ISO 802.3 (CSMA/CD), ISO 802.5 (TR)

Информация для заказа

Продукт	Номер изделия
NetWare FTAM v1.1	883-001361-001

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СЛУЖБ БАЗ ДАННЫХ

NetWare Btrieve

NetWare Btrieve — система управления записями, использующая индексацию по ключу и обеспечивающая высокую производительность работы с файлами для различных прикладных программ. Она также предназначена для использования другими производителями в качестве инструментальных средств, обеспечивающих функции управления записями низкого уровня, которые легко доступны любой прикладной программе.

NetWare Btrieve выполнена в виде дополнительного процесса (VAP) для серверов ELS Level I, Advanced и SFT NetWare и в виде загружаемого модуля NetWare (NLM) для серверов NetWare 386. Она состоит из двух программных модулей: Record Manager и Btrieve Requester. Record Manager выполняется на сервере и управляет всеми операциями доступа к файлам. Btrieve Requester выполняется на рабочих станциях и позволяет прикладным программам осуществлять прозрачную связь с Record Manager. Модули Btrieve Requester имеются для сред DOS, OS/2 и Windows. Данная конфигурация показана на рис.3.21.

Прикладные программы посылают запрос модулю Record Manager через Requester, использующий высокопроизводительные протоколы связи NetWare. Скорость обмена данными внутри сервера (а большая часть обменов происходит именно на сервере) ограничена только пропускной способностью его шин данных. Обмены данными, использующие низкоскоростную среду сетевого кабеля, ограничены только запросами клиентов и ответами сервера базы данных. Использование преимуществ архитектуры клиент/сервер, таким образом, позволяет Btrieve обеспечить передовую распределенную обработку в среде ЛВС. Btrieve также обеспечивает исключительную целостность данных, предлагая три метода контроля работы баз данных.

1. Btrieve использует систему "предварительной фиксации" для обеспечения защиты информации на уровне файлов при сбоях системы. Прежде чем изменить запись при выполнении транзакции, осуществляющей несколько изменений, Btrieve сохраняет "отпечаток" используемой области файла данных. Если сбой питания произойдет прежде, чем завершится операция изменения или транзакция, Btrieve автоматически восстановит файлы в то состояние, в котором они были до начала операции.
2. В сетях NetWare администраторы системы обычно задействуют систему отслеживания транзакций NetWare TTS, обеспечивающую защиту информации на уровне транзакций.
3. Для сохранения целостности данных на уровне нескольких транзакций Btrieve обеспечивает более

сложные возможности регистрации и “отката с повторением“. Btrieve регистрирует всю информацию о работе базы данных в контрольном файле. Если пользователь хочет восстановить файл данных Btrieve из старого архива, работа Btrieve, зарегистрированная в этом файле, может быть повторена с момента создания архива.

Другие средства защиты и обеспечения целостности включают блокировку на уровне записей, обработку транзакций, выявление тупиковых ситуаций, парольную защиту и шифрование.

Индексы и файлы динамически поддерживаются в таком состоянии, которое обеспечивает наиболее быстрый доступ к записям, даже если размер файла увеличивается. Записи могут быть переменной длины. Поддерживается отбрасывание пробелов и сжатие данных.

Вызывая подпрограммы Btrieve из своих прикладных программ, разработчики могут осуществлять последовательный или прямой поиск по значению ключа, а также вставлять, обновлять и удалять записи. Btrieve поддерживает прикладные программы предварительной обработки в средах DOS, OS/2 и Windows. Прикладные программы могут быть написаны на различных языках программирования, включая Си, BASIC, Pascal, Fortran, Modula-2, COBOL и APL.

Btrieve VAP встроен в NetWare v2.2. Btrieve NLM встроен в NetWare v3.11.

Версия Btrieve в виде NLM для NetWare 386 включает также интерфейсы, позволяющие другим NLM, выполняющимся на сервере, осуществлять доступ к Btrieve и файлам Btrieve. Поскольку Btrieve VAP является прикладной программой сервера NetWare, управление им может осуществляться, как любым другим компонентом сети NetWare. Стандартные процедуры управления и резервирования NetWare применимы к файлам Btrieve в той же мере, как и к другим файлам.

Версии Btrieve для сред DOS, OS/2 и Windows включены в инструментальный набор Btrieve Developer´s Kit.

Характеристики

- Включается в NetWare для обеспечения распределенной обработки.
- Устойчивость к сбоям системы.
- Защита с использованием паролей и шифрования информации.
- Поддержка рабочих станций DOS, OS/2 и Windows.
- Динамическая структура файлов и индексов.
- Сжатие последовательностей повторяющихся символов (дополнительно).
- Доступ по ключу или шаговые операции для физического прохождения по файлу..
- Простой программный интерфейс в виде набора подпрограмм.
- Поддержка Си, BASIC, Pascal, Fortran, Modula-2, COBOL, APL и других языков (включает библиотеки для более чем 25 компиляторов).
- Утилиты поддержки файлов и отладки.

Новые качества, имеющиеся в версии 5.10:

- Расширенные операции позволяют осуществлять поиск нескольких записей в результате одного запроса, вместо того, чтобы делать это по очереди. Может быть указан критерий ограничения для неключевого поля, позволяющий делать выборку участка записи, а не записи в целом.
- Операции контроля позволяют регистрировать все обращения к Btrieve. Утилита Roll Forward — “откат с повторением“ — позволяет администраторам сети выполнять все операции, записанные в контрольном файле, над файлами, восстановленными из архива. Таким образом можно восстановить файлы по состоянию на момент сбоя системы.
- Версия Btrieve в виде NLM для NetWare 386 использует многозадачные возможности NetWare v3.11 для одновременной обработки запросов рабочих станций на управление данными.

Реализация

NetWare Btrieve является встроенной частью операционной системы NetWare. Для NetWare v3.11 Btrieve выполнен в виде загружаемого модуля NetWare (NLM). Этот модуль Btrieve может быть загружен автоматически при начальной загрузке сервера. Он также может быть в любой момент загружен или выгружен командами LOAD и UNLOAD. Для NetWare v2.2 Btrieve выполнен в виде дополнительного процесса (VAP). Btrieve VAP может быть установлен при помощи утилиты, поставляемой вместе с этим продуктом, и загружаться при каждой начальной загрузке сервера.

Администраторы сети должны устанавливать и загружать NetWare Btrieve на сервер NetWare всегда, когда используются прикладные программы Btrieve. Если какая-либо прикладная программа получена от другого поставщика или разработана самими пользователями, администратору сети достаточно установить NetWare Btrieve на сервер. Заменив автономный Btrieve на рабочей станции на NetWare Btrieve Requester, пользователи получат возможность высокопроизводительной обработки прикладной программой клиент-сервера без необходимости модификации прикладной программы.

Необходимое аппаратное обеспечение

Btrieve VAP может работать на любой машине с процессором 286 или 386, поддерживаемой NetWare, включая IBM PC AT, PS/2 и совместимые ПЭВМ. NetWare v2.2 требует минимум 1 Мбайт ОЗУ. Для работы Btrieve VAP необходим еще 1 Мбайт ОЗУ. NetWare v3.11 требует минимум 2 Мбайт ОЗУ. Для работы Btrieve NLM необходимо дополнительно 256 Кбайт памяти.

На рабочей станции NetWare Btrieve может работать на IBM PC AT, PS/2 и совместимых ПЭВМ. DOS Requester требует 38 Кбайт памяти рабочей станции. DOS Requester и OS/2 Requester требуют от 30 до 80 Кбайт в зависимости от конфигурации.

Необходимое программное обеспечение

Btrieve VAP может работать в любой операционной системе NetWare, поддерживающей интерфейс VAP (ELS Level II, Advanced, и SFT NetWare версии 2.0 и выше). Btrieve NLM для NetWare 386 требует использования NetWare v3.x. DOS Requester — DOS версии 3.1 и выше, Windows Requester — Windows версии 2.0 и выше, а OS/2 Requester — OS/2 стандартной или расширенной редакции версии 1.1.

Опции

Все продукты баз данных Novell основаны на NetWare Btrieve. Xtrieve PLUS — это система поиска данных, использующая интерфейс меню и включающая гибкое и простое в использовании средство создания отчетов. Она позволяет пользователям каталогизировать, анализировать и обновлять информацию, хранящуюся в файлах Btrieve. NetWare SQL представляет собой высокопроизводительное ядро баз данных, разработанное Novell. Если система NetWare Btrieve загружена на сервер NetWare, NetWare SQL превращает Btrieve в мощное ядро SQL, используемое различными программами предварительной обработки. Более подробная информация содержится в разделе данного руководства, посвященном NetWare SQL.

Btrieve Developer´s Kits представляет собой API, использующийся для написания прикладных программ на основе Btrieve.

Спецификации

Максимальные ограничения Btrieve

Число записей в файле:	не ограничено
Размер файла:	4 Гбайта
Длина записи переменной длины:	64 Кбайт
Длина записи фиксированной длины:	90 байт
Длина ключа:	255 байт
Число индексов для одного файла:	24
Число открытых файлов:	не ограничено

Типы данных:

строка
L-строка
Z-строка
целый со знаком
с плавающей запятой
дата
время
десятичная дробь
деньги
с плавающей запятой (формат языка BASIC)
цифровая строка
логический
переменной длины
LVAR
с автоматическим увеличением
битовый

Атрибуты индекса:

двойной/уникальномодифицируемый/немодифицируемый
пустой
ручной
сегментированный
восходящий/нисходящий
с автоматическим увеличением

Интерфейсы языков программирования:

Microsoft C, COBOL, FORTRAN, QuickBASIC
IBM BASIC, COBOL, Pascal
Turbo BASIC, C, Pascal
Micro Focus COBOL, COBOL/2
Lattice C
Logitech Modula-2/86
MBP COBOL, COBOL-85
Realia COBOL
RM COBOL
STSC APL*PLUS
Novell Network C for DOS
Novell C Network Compiler/386
WATCOM C v8.0
WATCOM C 386 v7.0

Информация для заказа

NetWare Btrieve входит в комплект поставки NetWare версии 3.11 и выше и ELS, Advanced и SFT NetWare версии 2.0 и выше. За дополнительной информацией о продукте обращайтесь в отдел технической поддержки маркетинга фирмы Novell по телефону в США (512)794-1796 или в местное представительство фирмы Novell.

NetWare SQL

NetWare SQL — это высокопроизводительное ядро реляционных баз данных, разработанное Novell специ-

ально для операционных систем NetWare. Доступ к NetWare SQL могут осуществлять электронные таблицы, бухгалтерские пакеты, языки программирования четвертого поколения (4GL) и многие другие прикладные программы баз данных. Эти программы поддерживают рабочие станции DOS, OS/2 и Windows, позволяя администраторам сети создать надежную, высокопроизводительную систему распределенной обработки.

Продукт NetWare SQL разработан для оптимизации производительности локальной сети и прикладных программ. Благодаря тому, что большая часть операций обработки данных выполняется на сервере, трафик сети существенно сокращается. NetWare SQL также повышает целостность данных, поскольку такие задачи, как управление блокировкой записей, осуществляются преимущественно сервером, а не множеством рабочих станций в сети. Устойчивость к сбоям обеспечивается прозрачным использованием системы отслеживания транзакций TTS фирмы Novell.

NetWare SQL Requester реализован для сред DOS, OS/2 и Windows (как показано на рис.3.22). Для каждой среды имеются две версии Requester, одна из которых (требующая меньше памяти рабочей станции) выполняет преобразование и форматирование данных на сервере, а вторая (требующая больше памяти рабочей станции, но увеличивающая производительность прикладных программ) выполняет преобразование и форматирование данных на рабочей станции.

Прикладные программы посылают серверу запрос высокого уровня через NetWare SQL Requester, использующий высокопроизводительные сетевые протоколы связи. Использование преимуществ архитектуры клиент/сервер позволяет NetWare SQL Requester обеспечить эффективную распределенную обработку в среде ЛВС. Средства защиты и обеспечения целостности включают блокировку на уровне записей, обработку транзакций, выявление тупиковых ситуаций, парольную защиту и шифрование.

XQL представляет собой инструментальный набор для создания прикладных программ NetWare SQL. Интерфейсы прикладного программирования XQL поддерживают стандартные операторы SQL, а также расширения, обеспечивающие реляционные функции, выходящие за рамки SQL. (Более подробная информация содержится в разделе этого руководства, посвященном XQL.)

NetWare SQL обеспечивает распределенное управление базами данных, делая физическое расположение данных прозрачным для пользователей и прикладных программ. NetWare SQL поддерживает реляционные связи через несколько серверов, позволяя легко устанавливать отношения между отдельно хранящимися таблицами баз данных.

Поскольку NetWare SQL является прикладной программой сервера, управление ей может осуществляться так же, как любым другим компонентом сети NetWare. Стандартные процедуры управления и резервирования NetWare применимы к файлам базы данных NetWare SQL в той же мере, как и к другим файлам. Лицензия требуется для установки на каждый сервер.

Рис. 3.23. Архитектура NetWare SQL

Характеристики

- Позволяет различным прикладным программам одновременно использовать общую, хорошо защищенную базу данных.
- Поддерживает множество программ предварительной обработки, включая Concentric R&R, Lotus 1-2-3 v3.0, RaSQL фирмы Communication Horizons и SQL File фирмы Dave Vinzant.
- Повышает производительность прикладных программ и сети, реализуя архитектуру клиент-сервер.
- Повышает степень целостности и защиты информации за счет централизации основных функций.
- Обеспечивает распределенное управление базами данных и установление реляционных связей через несколько серверов.
- Улучшает доступ к ресурсам сервера и рабочих станций и эффективность их работы.
- Использует NetWare Btrieve для надежного, высокопроизводительного управления записями.
- Поддерживает открытый интерфейс прикладного программирования XQL, содержащий дополнительные реляционные функции.
- Поддерживает стандартный язык структурированных запросов (SQL).
- Поддерживает прикладные программы баз данных на рабочих станциях DOS, OS/2 и Windows.

Версия NLM

- Возможность динамического конфигурирования.
- Улучшенное управление памятью.
- Расширенные возможности.
- Может обеспечить функциями SQL другие прикладные программы и NLM, выполняющиеся на сервере.

- Существенное увеличение производительности при использовании NetWare 386.

Версии VAP и NLM

- Поддерживают препроцессор включаемых операторов SQL языка Си (в предыдущих версиях операторы SQL обрабатывались как строки ASCII).
- Хранят откомпилированные операторы SQL в словаре данных, что улучшает реакцию системы и сокращает трафик сети.
- Используют расширенные операции Btrieve Extended Operations для достижения максимальной производительности.

Реализация

NetWare SQL NLM, версия NetWare SQL в виде загружаемого модуля NetWare (NLM), был первым NLM, разработанным для операционной системы NetWare v3.x. NetWare SQL выполнен в виде VAP для серверов NetWare v2.x. Поскольку NetWare SQL позволяет интегрировать управление информацией в сети NetWare, администраторам сети стоит думать о покупке и установке NetWare SQL всякий раз, когда требуется высокопроизводительное SQL-ядро базы данных. Торговые агенты, консультанты вертикального рынка и разработчики программного обеспечения должны рассматривать возможность использования NetWare SQL всякий раз, когда им требуется создать для своих клиентов прикладные программы управления распределенными базами данных.

Поддержка промышленных программ предварительной обработки

Ниже следует список, содержащий некоторые прикладные программы других разработчиков, поддерживающие NetWare SQL, или программы, чьи изготовители объявили о том, что поддержка NetWare SQL будет обеспечена в течение 1991 года.

dANALYST Gold
Arago dBXL
Arago Quicksilver
Artemis Prestige
ENTERPRISE/WHO-WHAT-WHEN
Erwin/ERX
Forest & Trees
FormIT
Form WIN
Great Plains Accounting Software
Insight Report Writer
Intelligent Planner
Intelligent Query (IQ)
Lotus 1-2-3
Macola Software
Object Vision
Paradox
PdMain Software
Planet
Platinum Database Directionary
Production Data Management System
Q&E Server Database Edition
Q&E Database Library
Raison 10
RaSQL/X
ReportRight
R&R Report Writer for NetWare SQL
SQLtalk
SQL Windows
ToolBook
Visible Analyst Workbench
Visual Basic

Необходимое аппаратное обеспечение

Как уже говорилось, NetWare SQL выполняется на сервере NetWare. Из этого следует, что продукт может выполняться на любом поддерживаемом NetWare сервере. Версия VAP требует минимум 2 Мбайта, а версия NLM — 4 Мбайта ОЗУ.

NetWare SQL Requester реализован идентично в версиях NetWare SQL VAP и NML. DOS Requester, выполняющий преобразование данных на рабочей станции, использует 70 Кбайт ОЗУ; Requester сервера использует 30 Кбайт ОЗУ. Обе программы требуют использования DOS версии 3.1 и выше.

OS/2 Requester, выполняющий преобразование данных на рабочей станции, требует 40 Кбайт ОЗУ; версия Requester для сервера использует 90 Кбайт ОЗУ. Обе версии поддерживают OS/2 стандартной и расширенной редакции версии 1.1.

Windows Requester, выполняющий преобразование данных на рабочей станции, требует 40 Кбайт ОЗУ; версия Requester для сервера использует 90 Кбайт ОЗУ. Обе версии поддерживают Windows версии 2.0 и выше.

Необходимое программное обеспечение

Установленный на сервере NetWare SQL требует использования ELS Level II, Advanced или SFT NetWare версии 2.1 или выше, а также Btrieve VAP версии 5.10 или выше. NetWare SQL требует использования NetWare v3.11 и Btrieve NLM версии 5.10 или выше. Прикладные программы могут создаваться для среды DOS, OS/2 или Windows, поэтому тип и версия операционной системы рабочей станции зависит от конкретной прикладной программы.

Разработчику, создающему прикладные программы для NetWare SQL, потребуются совместимые продукты других поставщиков, например RaSQL, или поддерживаемые языки программирования (полный список при-

водится в пункте "Спецификации"), а также XQL — интерфейс прикладного программирования для NetWare SQL.

Опции

Все продукты баз данных Novell совместимы с NetWare SQL. Xtrieve PLUS представляет собой программу просмотра таблиц, использующую меню и включающую гибкое и простое в использовании средство создания отчетов. Она предназначена для работы с NetWare SQL и позволяет каталогизировать, осуществлять поиск, анализ и обновление информации, хранящейся в файлах Btrieve.

XQL представляет собой инструментальный набор для NetWare SQL. Программы, использующие интерфейс XQL, работают с сервером баз данных NetWare SQL, создавая прикладные программы клиент-сервера, оптимизированные для среды NetWare.

Модернизация

Недавно Novell выпустила версию NetWare SQL 2.11. Нынешние пользователи могут воспользоваться новыми средствами, имеющимися в версии 2.11, которые описаны выше в этом разделе.

Спецификации (на основе XQL API)

Максимальные ограничения NetWare SQL

Длина поля:	255 байт
Число полей в одном просмотре:	не ограничено
Число колонок в одном просмотре:	не ограничено
Число соединенных файлов в одном просмотре:	8
Число активных просмотров:	не ограничено
Число таблиц в базе данных:	не ограничено

Типы данных:

строка
L-строка
Z-строка
целый со знаком
с плавающей запятой
дата
время
десятичная дробь
деньги
с плавающей запятой (формат языка BASIC)
цифровая строка
логический
переменной длины
LVAR
с автоматическим увеличением
битовый
BLOB

Характеристики файлов изложены в описании продукта Btrieve.

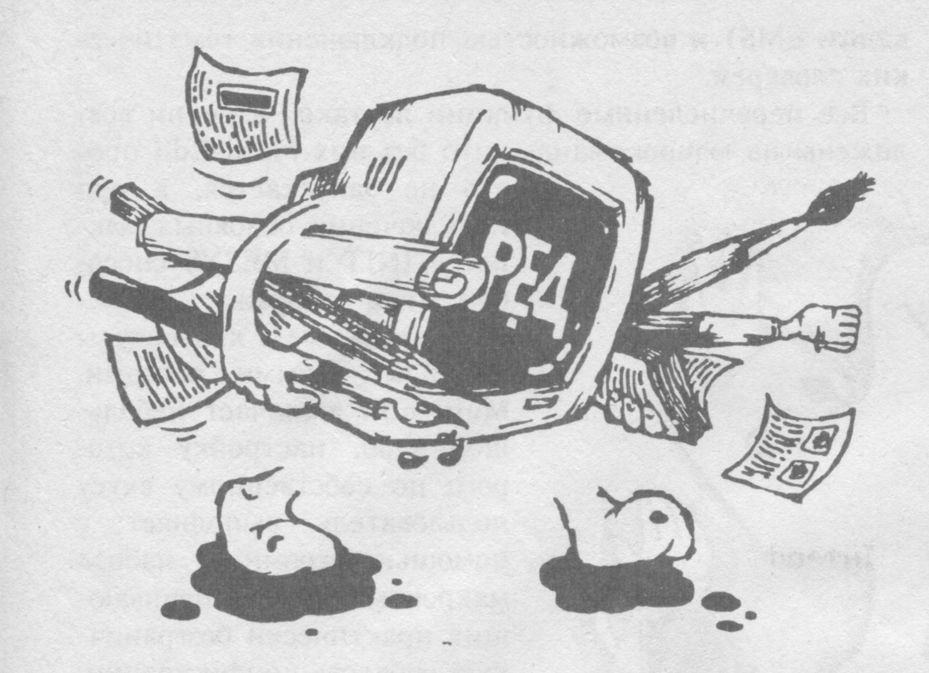
Интерфейс языков программирования:

BASIC
Lattice C
Microsoft CMicrosoft BASIC
Microsoft Pascal
Microsoft QuickBASIC
Micro Focus COBOL/2
Realia COBOL
Turbo C
Turbo Pascal
MBP COBOL
WATCOM C v8.0
Novell C for NLMs
C++ фирмы Borland
WATCOM C 386 v7.0

Информация для заказа

NetWare SQL продается уполномоченными торговыми агентами Novell. За дополнительной информацией о продукте обращайтесь в отдел технической поддержки маркетинга фирмы Novell по телефону в США (512)794-1796, или в местное представительство фирмы Novell.

Продукт	Номер изделия
NetWare SQL v2.11 (версия VAP: лицензия на 1 сервер)	
5.25-inch diskettes	883-000710-001
3.5-inch diskettes	883-000711-001
VAP upgrade from v2.01	
5.25-inch diskettes	883-001041-001
3.5-inch diskettes	883-001042-001
NetWare SQL 386 v2.11a (версия NLM: лицензия на 1 сервер)	
5.25-inch diskettes	883-000957-001
3.5-inch diskettes	883-000958-001



Предлагаемая вашему вниманию статья посвящена одному из наиболее популярных текстовых редакторов — пакету Multi-Edit компании American Cybernetics Inc. Не во всем соглашаясь с мнением автора, редакция КомпьютерПресс все же отдает должное той высокой степени профессионализма, с которой он описывает сильные стороны пакета.

Multi-Edit 5.00 — обзор основных возможностей

В настоящее время стандартом "de facto" в России стали IBM PC/AT-совместимые компьютеры, большую часть которых составляют 12-мегагерцевые с 1Мбайтом оперативной памяти, винчестером на 40 Мбайт и достаточно медленным видеоадаптером, что, вообще говоря, с натяжкой можно назвать рабочим инструментом программиста, а с учетом цен на современном рынке компьютерной техники вряд ли следует ожидать быстрого обновления парка персональных компьютеров. Поэтому разработчикам программного обеспечения зачастую приходится самостоятельно создавать инструментальные средства или использовать продукты своих коллег, оптимизируя скорость их выполнения при максимальной полезной нагрузке. Таким образом, на диске появляется огромное число безусловно нужных, но разрозненных программ, на написание и освоение которых и уходит наше драгоценное время.

Что ни говори, а основную часть своего рабочего времени программист-профессионал проводит за написанием или анализом программ. И то, и другое требует использования некоторого редактора. Так что для обращения к большинству утилит надо либо использовать резиденты, что урезает и без того скудные ресурсы памяти, либо всякий раз выходить из редактора в лучшем случае в DOS SHELL. Все эти проблемы давно решены при помощи интегрированных программных сред, на нижней ступени которых стоят интегрированные среды компиляторов, а на верхней — многозадачные системы типа Windows. Но первые слишком узко специализированы, а вторые слишком мощны и, подчас, слишком избыточны, чтобы эффективно работать на компьютерах с низкой производительностью.

Приятно выделяется в этом отношении редактор Multi-Edit 5.00 Professional американской фирмы American Cybernetics Inc., который, на мой взгляд, обладает всеми возможностями для разработки интегрированной среды, как можно полнее отвечающей нуждам пользователя.

Обычно все обзоры текстовых редакторов сводятся к оценке их с точки зрения удобства подготовки текстов документов, но не следует забывать, что программа — тоже текст и удобство ее написания играет не последнюю роль на этапе разработки собственно программного продукта, а на этапе анализа приобретает просто первостепенное значение. Можно возразить, что большинство компиляторов имеет удобные интегрированные среды, в состав которых входит и редактор, и отладчик, и минимальный набор необходимых утилит. И все же не следует торопиться с выводами. Как быть с такими языками программирования, как Assembler или Clipper Summer´87, ведь они не имеют интегрированных сред. Кроме того, мне не раз приходилось на-

блюдать, что программисты достаточно высокого уровня на определенном этапе предпочитают работать со строчными компиляторами даже в таких мощных пакетах, как Turbo Pascal и C++, а для внесения изменений в код используют встроенный редактор Norton Commander. Они не только выигрывают в скорости создания результирующего модуля из исходного текста, но и получают необходимый доступ к большому количеству собственных утилит, используя при этом аппарат настройки на расширение пакета Norton Commander и возможности его пользовательского меню.



Итак, приведенные выше рассуждения позволяют выявить два критерия, которыми, вероятно, руководствуются все разработчики:

1. Скорость получения исполняемого кода.
2. Доступность необходимого набора утилит, используемых при написании и анализе программ.

Из личного опыта могу добавить, что доступ к документации по MS-DOS, краткому руководству по функциям и процедурам используемого языка тоже играют не последнюю роль.

Рассмотрим Multi-Edit с точки зрения выполнения этих требований, но для начала сделаем небольшое отступление и кратко остановимся на принципах построения Multi-Edit как системы.

В ряду текстовых процессоров Multi-Edit занимает несколько обособленное положение. Это объясняется, в основном, наличием мощного языка макрокоманд и аппаратной логикой построения системы. Под аппаратами понимаются наборы процедур и функций интерпретируемого макроязыка, которые обеспечивают:

- поддержку оконной организации системы;
- организацию файловой системы;
- полную поддержку мыши;
- продуманную систему HELP;
- систему меню стандарта WINDOWS\SAA;
- набор низкоуровневых функций, позволяющих получить доступ в макроязыке к системе прерываний DOS/BIOS (кроме портов ввода/вывода);
- и, наконец, средства для собственно обработки текста, работы с блоками, операций поиска и замены, конфигурирования экрана, поддержки EMS, программирования COM-порта и т.д.

Пакет предоставляет пользователю некоторый набор неизменных функций, гибкое использование возможностей которых позволяет добиваться желаемых результатов. Так например, использование встроенного аппарата SPELL CHECKER и HELP позволило организовать англо-русский словарь на 16,5 тыс. слов со средним временем доступа 3-4 секунды (при использовании EMS) и возможностью подключения тематических словарей.

Все перечисленные функции до такой степени возложены на макрокоманды, что без них Multi-Edit просто не запускается, а при подключении основных макросов INIT и MESYS способен лишь отображать на экране вводимые с клавиатуры символы. Иными словами, Multi-Edit включает небольшое ядро, настройку которого по собственному вкусу пользователь выполняет с помощью огромного набора макрокоманд, обеспечивающих практически безграничную гибкость конфигурации, да еще в такой "интимной" области, как подготовка текстов и написание программ. А вы видели когда-нибудь пользователя, полностью удовлетворенного возможностями систем такого класса? Возьму на себя смелость высказать предположение — если Clipper является идеальной системой для организации баз данных, то Multi-Edit является аналогичной по возможностям системой обработки текстов.

Теперь вернемся к проблеме эффективности среды Multi-Edit при написании программ. Как уже упоминалось, одним из основных критериев служит скорость получения исполняемого модуля. На низкоскоростных компьютерах она существенно зависит от того, как распорядиться оставшейся памятью и сконфигурировать дополнительные 384 Кбайта. Одним из основных достоинств Multi-Edit является хорошо продуманный алгоритм и реализация свопинга. Это позволяет быстро выходить в DOS и выполнять необходимые утилиты. При этом резидентно остается только 1.5 Кбайта, которые необходимы для восстановления системы. Тем самым освобождается почти вся память для нужд компилятора и/или программ пользователя. После выполнения программы происходит полное восстановление состояния редактора для всех загруженных в него файлов, при этом сохраняются размеры окон, положение курсора, отметка блоков и др. Процессом свопинга можно управлять во время работы, используя соответствующий аппарат поддержки EMS (если она подключена).

За счет быстрого свопинга обеспечивается и возможность обработки файлов большого размера — типично узкого места в большинстве редакторов. Пакет вполне удовлетворительно решает, например, проблему работы с листингами языка Assembler. В системе можно одновременно держать до 100 файлов, что позволяет работать с большим количеством модулей при структурном программировании и разработке сложных систем. Каждое окно имеет индивидуальную

настройку: размеры, положение курсора, размещение табуляторов и т.п. Кроме того, текст в окне может отображаться двумя цветами — цвет измененных со времени загрузки файла строк отличается от основного текста, что существенно облегчает процесс анализа программы на этапе внесения в нее необходимых изменений.

Теперь о проблеме утилит. В своем большинстве они написаны на макроязыке, в связи с чем на их модификацию уходит меньше времени, так как можно пользоваться стандартными функциями, например, готовой системой меню, а благодаря относительной компактности макрокода на хранение тратится меньше дисковой памяти. Макросы обладают достаточно быстрой скоростью выполнения, причем, указав тип макроса при компиляции, можно обеспечить ему своего рода резидентность, тем самым увеличив скорость его выполнения. И, что не менее важно, макросы можно привязать к постоянным клавишам или комбинациям клавиш, обеспечив тем самым унификацию пользовательского интерфейса. Ведь, работая в различных средах, пользователю каждый раз приходится запоминать раскладку функций на клавиатуре, а она часто бывает не самой удобной.

Макроязык Multi-Edit паскалеподобен. Он обеспечивает достаточно мощные средства для разработки системных утилит. Это продемонстрировал, например, в своих разработках Алексей Романов из ВЦ Москов-

ского Университета (см. Софтпанораму № 35 за 1991 год). Им разработаны макросы просмотра дампа памяти, таблицы векторов прерываний, шестнадцатиричный вьюер двоичных файлов, написана система, позволяющая включать в макрокод исполняемые ассемблерные вставки, то есть получать "шитый" макрокод. Перечисленного вполне хватает на "джентльменский набор" утилит системного программиста.

Теперь немного о собственных разработках.

1. Поскольку наиболее привычной файловой системой все же остается Norton Commander, мне показалось интересным и полезным эмулировать средствами макроязыка основные возможности этого пакета в области работы с файлами. Дополнительно были реализованы интерфейс с основными архиваторами, полное управление очередью фоновой печати, буферизация файлов и имен часто используемых каталогов, настраиваемый план клавиш. Такая совокупность макросов позволила достаточно эффективно работать с файловой системой, практически не выходя из редактора.

2. У программистов часто возникает необходимость рассчитать экранную форму, то есть вывести координаты основных элементов формы с целью их дальнейшего использования в программе. Разработчикам программ на Clipper, вероятно, знакома система разработки экранных форм UI, занимающая на диске 250 Кбайт (вторая версия занимает около 600 Кбайт). Аналогичный комплект утилит, разработанный на макроязыке и позволяющий выполнять необходимый набор этих функций и затем генерировать исходный текст программы на нужном языке, занимает не более 20 Кбайт.

3. Настройка системы производится теперь не только по расширению редактируемых файлов, но и по языку программирования, что позволяет, при использовании системы получения скелета программы, облегчить анализ ее структуры и быстро устанавливать курсор при просмотре на ключевые конструкции (вызовы процедур и функций, метки и т.п.).

4. Реализован аппарат обработки ошибок компиляции, эмулирующий обработчик ошибок Turbo Pascal 6.00 Borland International.

Приведенные примеры убедительно доказывают, что с помощью макроязыка можно реализовать большинство необходимых для работы вспомогательных программ, то есть создать интегрированную настраиваемую оболочку теоретически для любого языка программирования.

Преимуществом такого подхода будет являться универсальность, интегрируемость и взаимосвязь системы утилит, что позволит исключить дублирование функций и связать их возможности с особенностями создаваемого текста программы. Конечно, такой подход имеет и свои недостатки — все написанные утилиты перестают быть автономными и работают только под управлением Multi-Edit. Но в сочетании с возможностью расширения среды и существенными трудностями, связанными со сменой компьютера на более мощный, когда можно будет использовать многофункциональные интегрированные пакеты типа WINDOWS 3.0, это не такой уж большой недостаток.

Теперь об организации системы HELP. Одной из основных особенностей интегрированных программных сред является мощная система помощи Hyper Help, обеспечивающая как получение справочной информации по самим функциям среды, так и вывод контекстно зависимых подсказок по функциям и процедурам используемого языка программирования. Система Hyper HELP Multi-Edit позволяет организовывать HELP-файлы практически любого размера со сквозной системой ссылок как внутри одного файла, так и среди нескольких HELP-файлов. Hyper HELP позволяет выделять текст в файле шестью цветами, динамически подзагружать HELP-файлы и даже выполнять модули после выбора очередной ссылки по тексту. Иными словами, система HELP сравнима по возможностям с системой Norton Guide, что вполне соответствует современным требованиям. Хотя в стандартном макросе Multi-Edit не предусмотрена возможность контекстно зависимой подсказки, небольшие доработки позволили ликвидировать этот недостаток и теперь, при соблюдении некоторых минимальных правил организации HELP-файла, можно организовать такую возможность для любого языка. В настоящий момент реализована поддержка для Clipper´87, Tasm 2.00 и собственно макроязыка Multi-Edit. HELP-система обеспечивает возможность вырезания и приклеивания текста на экране, а простота процесса подключения переведенной на русский язык документации делает ее еще более привлекательной.

Кроме всего прочего в Multi-Edit предусмотрена возможность автоматического сохранения файлов, создание резервных .BAK-копий, мощный аппарат отката и сохранение полного состояния системы на диске между сеансами работы, скорость которого, после изменения алгоритма сохранения, становится вполне приемлемой.

Multi-Edit является не просто редактором, он обладает всеми необходимыми возможностями для написания собственной интегрированной среды, которая наиболее полно будет отвечать требованиям пользователя. Пакет Professional включает до 1.5 Мбайт исходного кода всех (за исключением MEDEBUG.MAC) макросов, что позволяет не только ознакомиться с "кухней" системы, но и исключить ненужные функции или заменить их более эффективными. Именно так была решена проблема обработки кириллицы.

Теперь хочется кратко остановиться на описании возможностей основных аппаратов Multi-Edit, без чего обзор продукта фирмы American Cybernetics Inc. будет явно неполным.

МЕНЮ. Позволяет организовывать меню любой сложности и конфигурации с неограниченным уровнем вложенности. Поддерживаются горизонтальные и вертикальные меню с практически неограниченным числом альтернатив, боксы ввода данных, элементами

которых служат поля 15 типов, в том числе поля-списки и поля — списки файлов. Вся система меню реализована одной макрокомандой USERIN.MAC. В пакет поставки входит исходный текст. К системе меню относится и несложный менеджер баз данных.

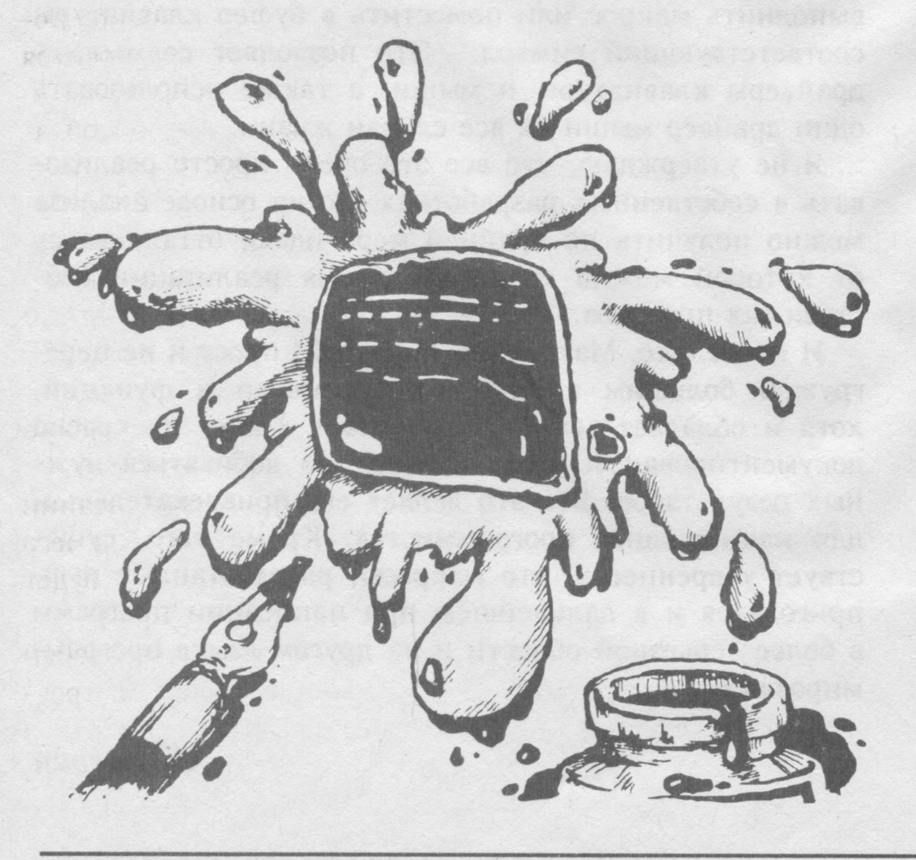
ОКОННАЯ СИСТЕМА. Позволяет открывать до 100 окон, оперативно переходить из одного окна в другое в процессе работы, копировать и перемещать блоки из окна в окно, динамически изменять размер и основные параметры окон. Имеется также и поддержка функций clipboard, аналогичная ID Borland International.

БЛОКИ. Поддерживается три вида блоков: строчный, прямоугольный и поточный. Блоки текста можно копировать, удалять, перемещать и сохранять на диске в виде отдельного файла, а также восстанавливать их с диска. Небольшие доработки позволяют перенести функции отметки блоков на привычную комбинацию «Shift+Arrow_Keys».

ФАЙЛЕР. Позволяет выполнять практически все операции с файлами. Можно одновременно работать с четырьмя панелями, отмечать и сортировать списки файлов, быстро получать исчерпывающую информацию о файле.

ПОДДЕРЖКА МЫШИ. Хорошо продуманная система, позволяющая работать со всеми элементами экрана. Нажатие кнопок мыши рассматривается как некоторое событие, информация о котором фиксируется во внутренних переменных системы, а поэтому может быть просто и эффективно обработана. Каждое окно снабжено дополнительной правой и нижней линейкой прокрутки, что позволяет быстро перемещать по экрану текст документа при помощи мыши.

КОНФИГУРАЦИЯ ЭКРАНА. Представлена пятью основными элементами: строкой состояния системы, строкой сообщений, строкой меню (по желанию пользователя), собственно рабочим полем и строкой экспресс-справки по функциональным клавишам. Каждый из элементов состоит из нескольких частей и может конфигурироваться отдельно или вовсе отсутствовать на экране. Все элементы системы меню и экрана описываются 50 различными цветами.



АППАРАТ ПОИСКА и ЗАМЕНЫ. Мощный и эффективный инструмент для работы с текстом. Выражения-шаблоны (regular expression) обеспечивают обработку "невидимых" символов, таких, как "начало" или "конец строки", специальных множеств или "классов" символов — строк неизвестного состава, объединяющих группы символов по одному образцу для поиска и/или замены. Имеется аппарат комплексного файлового поиска. Он обеспечивает сканирование указанного файла до первого вхождения поискового аргумента с выдачей номера соответствующей строки. Эту возможность можно расширить соответствующим "несанкционированным" использованием поисковых функций аппаратов HELP и SPELL CHECKER, которые позволяют возобновить поиск.

ОПЕРАЦИИ ТЕКСТОВОГО ПРОЦЕССОРА. Позволяют обрабатывать собственно текст в окне — форматировать, выделять и анализировать части текста. Аппарат представлен небольшим и самым необходимым количеством функций обработки строк и управления табуляцией, процедурами позиционирования в тексте документа и обширным набором внутренних переменных, которые дают исчерпывающее представление о настройке системы на редактируемый документ и помогают быстро и эффективно производить его обработку.

COM-ПОРТ. Имеется возможность программирования и анализа текущего состояния COM-порта. Обеспечивается возможность приема и передачи символов, встроенная поддержка протоколов XMODEM и YMODEM, а с внешней программой и ZMODEM. Также эмулируется ASCII-терминал. Все это позволяет создавать внутри системы необходимые коммуникационные утилиты.

Помимо описанных, существуют аппараты для управления свопингом, запуском внешних программ под управлением Multi-Edit и анализом их завершения, организации службы времени, мощная система Hyper Help и английский Spell Checker (впрочем, подключить русский синтаксический анализатор, например RUSP, не составляет особого труда), аппарат управления печатью и многое другое.

Коротко об особенностях макроязыка. Как уже упоминалось, он паскалеподобен, в нем используется 3 типа локальных переменных:

- строчный (символьный) тип. Поддерживаются операции слияния-конкатенации и сравнения строк;
- целый (и логический 0/1) тип. Имеется возможность представления целых чисел в шестнадцатиричном представлении. Поддерживаются все арифметические операции, а также автоинкремент — "++" и автодекремент — "--" для локальных переменных, все операции сравнения и операции "xor", "or" и "and";

- действительный (операции с плавающей точкой) тип. Поддерживаются арифметические операции и операции сравнения действительных чисел.

Имеются функции преобразования из одного типа в другой. Глобальные переменные, доступные любой макрокоманде в момент работы, могут относиться к одному из двух типов — INT или STR. Базовые языковые конструкции взяты из Pascal:
- if elsif end
- do while
- goto «Label»
- Call «Sub_Program_Name»

Передача параметров между макросами реализована при помощи внутренней переменной MPAPM_STR со стандартными процедурами ее разбора, а возврат — через две системные переменные — RETURN_STR и RETURN_INT, либо через глобальные переменные. Массивов в макроязыке нет, но они достаточно просто эмулируются при помощи индексируемых глобальных переменных.

Получаемый в результате компиляции макрокод интерпретируется системой после выдачи команды RUN_MACRO(´Имя_Макро´). Загрузкой и удалением макрокоманд можно управлять при помощи других макросов или путем указания их типов в процессе компиляции: NOSWAP (резидентно), TRANS (транзитно), DUMP (полная выгрузка) и т.п.

Предусмотрена неплохая система обработки ошибок путем анализа переменной кода ошибки — ERROR_LEVEL, причем если такой анализ не был предусмотрен программистом, Multi-Edit сам корректно распознает ее и обрабатывает.

Хочется остановиться еще на двух моментах. Во-первых, на мой взгляд достойна внимания логика организации и подход, предлагаемый в Multi-Edit для решения традиционных задач, возникающих при разработке сложных систем и интегрированных сред, например, при написании сложных программ, с интерфейсом в виде системы меню и управляющих клавиш (hotkeys). Даже при программировании на одном языке не сразу удается организовать систему так, чтобы она была универсальна и подходила для решения хотя бы 80% задач. Конечно, существуют готовые решения и некоторые стандарты на стиль интерфейса, которые предлагают фирмы-разработчики, поставляющие соответствующие средства в составе пакета даже в виде исходных модулей. Но они не всегда эффективны, а иногда и слишком громоздки. После некоторых экспериментов в этой области я остановился на подходе, который предлагается в Multi-Edit для решения проблемы — каждое меню, за исключением горизонтальных и вертикальных меню с фиксированным количеством альтернатив, представимо некоторым множеством полей пятнадцати различных типов. На основе описания этих полей отдельный модуль (в случае Multi-Edit это макрокоманда DATA_IN) выполняет построение меню, перемещение курсора между опциями и вывод оперативной подсказки по использованию полей каждого типа. Результат возвращается в виде двух глобальных переменных символьного и целого типа. А произведенные пользователем установки — в виде массивов. Для Multi-Edit это соответствует переменным Return_Str, Return_Int и индексируемым глобальным перменным, заменяющим массивы. Наличие исходного кода макро USERIN позволило достаточно легко разобраться в идеологии решения задачи.

Во-вторых, заслуживает внимания система организации HELP. Приведена практически вся необходимая логика для организации аналогичной системы средствами другого языка программирования. Для процедуры back-track — “возврат к предыдущему разделу“ — использован стековый механизм, а выделение и ссылки на фрагменты текста организованы в виде системы спецсимволов.

Вообще говоря, Multi-Edit достаточно интенсивно использует преимущества стека — так организован аппарат отображения меток функциональной клавиатуры, механизм отката, маркировки и т.п. Еще одна особенность — handle-события. С их помощью организовано управление таймером и, частично, мышью.

В первом случае с некоторым handle, который помещается в стек LIFO, связывается макрос. Система с частотой одного тика таймера производит опрос стека и обращается к макросу, на который установлен handle, а он, в свою очередь, используя аппарат анализа событий таймера, может передать управление дальше по стеку, выполниться и передать управление, или, наконец, вывести событие из стека процедурой KILL_TIMER_EVENT(handle).

Во втором случае каждое нажатие кнопки мыши рассматривается как событие (event), которое должно быть заранее описано. Описывается местоположение элемента на экране (например, “Cancel“ на нижней границе активного меню) и действия, которые необходимо выполнить при наступлении этого события — выполнить макрос или поместить в буфер клавиатуры соответствующий символ. Это позволяет совместить драйверы клавиатуры и мыши, а также использовать один драйвер мыши на все случаи жизни.

Я не утверждаю, что все это очень просто реализовать в собственных разработках, но на основе анализа можно получить по крайней мере идею, отталкиваясь от которой можно сократить время реализации собственных проектов.

И последнее. Макроязык Multi-Edit прост и не перегружен большим количеством процедур и функций, хотя и обладает всем необходимым. Пакет прекрасно документирован и позволяет быстро добиваться нужных результатов. Все это делает его привлекательным для начинающих программистов. Кроме того, существует уверенность, что макросы, разработанные ими, пригодятся и в дальнейшем при написании программ в более серьезной области и на другом языке программирования.

О. Аверин



Мы продолжаем публикацию фрагментов из книги Д.Рогаткина и А.Федорова "Turbo Pascal в среде Windows".

Компилятор Turbo Pascal for Windows

Эта книга посвящена написанию Windows-программ с использованием программного продукта фирмы Borland — компилятора Turbo Pascal for Windows (TPW), обладающего рядом уникальных возможностей. Во-первых, Turbo Pascal for Windows — первый компилятор языка Pascal, специально разработанный для Windows. Во-вторых, он позволяет создавать программы, использующие технологию объектно-ориентированного программирования. В-третьих, в комплекте с компилятором поставляется объектно-ориентированная библиотека классов **Object Windows**, использование которой существенно упрощает создание Windows-программ. И наконец, программисты, ранее использовавшие DOS-версии компилятора Turbo Pascal, смогут теперь без особых затрат перейти к созданию Windows-программ.

Наряду с **Object Windows** и собственно компилятором в составе пакета Turbo Pascal for Windows поставляются встроенный ассемблер **BASM**, модуль **Strings**, предназначенный для обработки строк формата языка Pascal и ASCIIZ-строк, модуль **WinCrt**, используемый для быстрого преобразования DOS-программ в Windows-программы, отладчик Windows-программ **Turbo Debugger for Windows** и многое другое.

Прежде чем рассматривать различные аспекты программирования в среде Windows (чему и посвящена предлагаемая вам книга), остановимся на описании самого компилятора и включенных в него средств.

Интегрированная среда разработчика

Наиболее впечатляющая часть компилятора Turbo Pascal for Windows — интегрированная среда разработчика (ИСР). В отличие от компиляторов языка С фирм Microsoft, Borland и Zortech, также используемых для разработки Windows-программ, весь процесс создания программы — ввод исходного текста, редактирование, компиляция, выполнение и отладка — проходит непосредственно в среде Windows. ИСР практически идентична среде компилятора Turbo Pascal 6.0 по системе команд. Она позволяет открывать при разработке сложных программных комплексов до 32 окон, каждое из которых может быть затем максимизировано или минимизировано (т.е. превращено в иконку).

Текстовый редактор поддерживает систему команд стандарта SAA/CUA фирмы IBM и систему команд редактора Wordstar. Использование редактора Wordstar покажется особенно удобным тем программистам, которые ранее уже работали с DOS-версией компилятора Turbo Pascal. Справочная система типа гипертекст содержит информацию по всем аспектам использования компилятора — от конструкций языка и стандартных библиотек до библиотеки **Object Windows** и функций **Windows API**.

Пакетный компилятор

Помимо компилятора, работающего в составе ИСР, в состав поставки входит пакетный компилятор TPCW.EXE, позволяющий создавать Windows-программы вне среды Windows. Опции этого компилятора идентичны командам меню компилятора, работающего в составе среды. Полный список опций пакетного компилятора приводится в приложении.

Для компиляции программы с помощью TPCW.EXE используется следующая командная строка:

```
tpcw winprg.pas
```

Данные о расположении стандартных модулей и используемых опциях компилятора находятся в файле *TPCW.CFG*.

При запуске пакетного компилятора без указания имени компилируемого файла на экран выводится список допустимых опций командной строки:

```
Turbo Pascal for Windows
Copyright (c) 1991 Borland International

Syntax: TPCW [options] filename [options]
/B       Build all units          /$A-    No word alignment
/Dxxx    Define conditionals      /$B+    Complete boolean
                                          evaluation
/Exxx    EXE & TPU directory      /$D-    No debug information
/Fxxx    Find run-time error      /$F+    Force FAR calls
/GS      Map file with segments   /$G+    Generate 286
                                          instructions
/GP      Map file with publics    /$I-    No I/O checking
/GD      Detailed map file        /$L-    No local debug
                                          symbols
/Ixxx    Include directories      /$Mxxx  Memory allocation
                                          parameters
/L       Link buffer on disk      /$N+    8087 code generation
/M       Make modified units      /$R+    Range checking
/Oxxx    Object directories       /$S-    No stack checking
/Q       Quiet compile            /$V-    No var-string
                                          checking
/Rxxx    Resource directories     /$W-    No Windows stack
                                          frames
/Txxx    Turbo directories        /$X-    No extended syntax
/Uxxx    Unit directories
/V       EXE debug information
```

Средство создания ресурсов

Windows-программы помимо кода могут содержать специальный тип данных — ресурсы, в которых хранятся элементы интерфейса Windows-программы — иконки, меню, панели диалога, курсоры и т.п. Ресурсы располагаются специальным образом внутри выполняемого файла. Создавать и редактировать ресурсы можно при помощи входящего в состав TPW редактора ресурсов **Whitewater Resource Toolkit (WRT)**, первоначально разработанного фирмой Whitewater Group для языка Actor. Этот редактор имеет средства для редактирования всех типов ресурсов в рамках единой среды Resource Manager. С помощью WRT можно создавать и редактировать практически все типы ресурсов, используемых Windows-программами, работая с файлами типа .RES (файл ресурсов), .EXE (выполняемый файл) и .DLL (динамически загружаемая библиотека).

Whitewater Resource Toolkit позволяет создавать следующие типы ресурсов: описатели командных клавиш, графические изображения (*.BMP*), курсоры (*.CUR*), панели диалога, иконки (*.ICO*), меню и строковые ресурсы. Создаваемые и редактируемые ресурсы могут быть сохранены либо в текстовом виде, для их последующей компиляции с помощью компилятора ресурсов **RC**, либо в виде, пригодном для непосредственного подсоединения к выполняемым файлам.

Отметим, что WRT не позволяет редактировать шрифты, файлы данных в формате RC и нестандартные (определяемые пользователем) ресурсы.

При необходимости ресурсы могут быть подключены к программе при помощи директивы **$R**. В этом случае объем подключенных к программе ресурсов в двоичном виде не должен превышать 64 Кбайт. Для подключения ресурсов большего размера необходимо использование редактора ресурсов.[1]

Отладчик Windows-программ

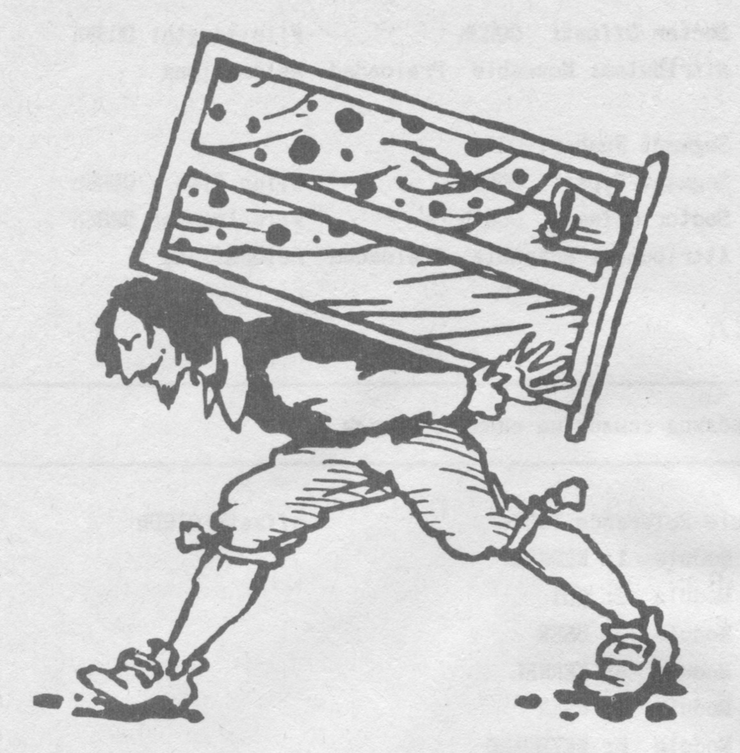
В отличие от предыдущих версий компилятора (4.0 и выше), Turbo Pascal for Windows включает в себя только внешний отладчик, вызываемый из ИСР командой **Run:Debugger**. Отладчик **Turbo Debugger for Windows** работает как полноэкранная DOS-задача и при внешнем сходстве со своей DOS-версией обладает рядом дополнительных возможностей, необходимых для отладки Windows-программ:
- просмотр Windows-сообщений;
- просмотр содержимого локальной и глобальной кучи;
- просмотр модулей, загруженных Windows;
- отладка динамически загружаемых библиотек (DLL).

В данной версии отладчика (2.5) используется приобретенная по лицензии у фирмы Microsoft динамическая библиотека *WINDEBUG.DLL*, которая также используется в отладчике **CodeView** и предназначена только для работы в текстовом режиме. Работа в текстовом режиме накладывает ограничения на работу Windows: отладчик сохраняет графическое изображение на экране только в том случае, если используется стандартный драйвер видеоадаптера. Например, невозможно восстановление экрана при работе в режиме 256 цветов с разрешением 800x600 точек с видеоадаптером VGA Video7, Paradise VGA и т.п. Хочется надеяться, что этот существенный недостаток будет со временем устранен. Turbo Debugger for Windows позволяет выполнять отладку Windows-программ на двух мониторах, удаленную отладку через COM-порт, а также отладку с использованием аппаратных прерываний.[2]

---

[1] Вскоре после реализации компилятора Turbo Pascal for Windows фирма Borland выпустила собственную версию редактора ресурсов — **Resource Workshop (RW)**. Редактор ресурсов Resource Workshop поставляется в составе версии компилятора Turbo Pascal for Windows 1.5.

[2] Входящая в комплект компилятора Borland C++ 3.0 версия отладчика для Windows-программ Turbo Debugger for Windows 3.0 страдает тем же недостатком, что и этот пакет, но в данной версии предпринята попытка вынести работу с видеоадаптером во внешнюю библиотеку, что делает возможным создание пользовательских библиотек для других типов видеоадаптеров. Поставляются библиотеки для видеоадаптеров фирм TSENG Labs — TSeng ET-3000/ET-4000 (файл TSENG.DLL) и ATI EGA WONDER (файл ATI.DLL).

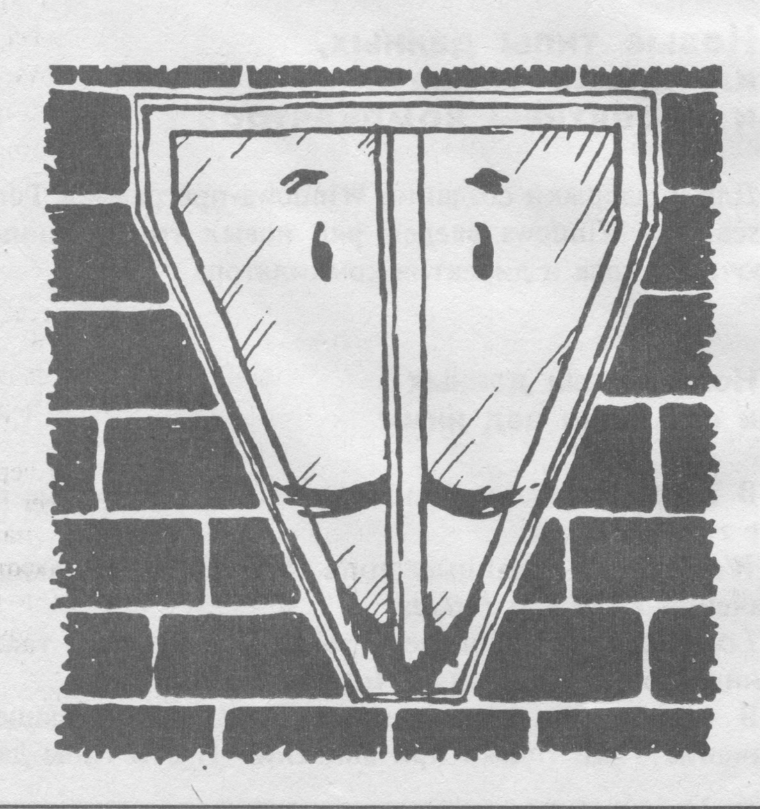


Утилиты

В комплект поставки входит также несколько утилит, которые могут оказаться полезными при разработке Windows-программ.

HC.EXE

Компилятор создания справочной системы Windows-программ. Создание справочной системы — довольно сложный вопрос, подтверждением чему может служить, например, наличие целого тома документации на эту тему. Вместе с компилятором поставляются исходный текст небольшой справочной системы и пример ее использования в прикладной программе. Компилятор HC разработан фирмой Microsoft и поставляется по лицензии.



RC.EXE

Пакетный компилятор ресурсов может быть использован для создания ресурсов в виде текстовых файлов. В большинстве случаев использование редакторов ресурсов WRT или RW оказывается более удобным.

TPUMOVER.EXE

Эта утилита может использоваться при просмотре содержимого стандартной библиотеки модулей **TPW.TPL**, включения в нее новых модулей и извлечения существующих. Если утилита запускается без опций, то на экран выводится содержимое указанного библиотечного модуля:

```
TPU Mover  Version 6.00
Copyright (c) 1991 Borland International

Unit          Code  Data  Uses
SYSTEM       11582   604
STRINGS        960     0  SYSTEM
WINTYPES         0     0  SYSTEM
WINPROCS       196     0  SYSTEM WINTYPES
WINDOS        1915     2  SYSTEM WINTYPES WINPROCS STRINGS
WINCRT        4066   338  SYSTEM WINTYPES WINPROCS WINDOS
```

При вызове утилиты используются следующие командные строки:

- для извлечения существующих модулей из стандартной библиотеки:

```
tpumover tpw.tpl *windos
```

- для включения новых модулей:

```
tpumover tpw.tpl +myunit.tpu
```

- для удаления модуля из библиотеки:

```
tpumover tpw.tpl -myunit
```

Внимание: При удалении модулей из библиотеки необходимо помнить, что утилита TPUMOVER не создает резервной копии библиотеки и не выдает предупреждающих сообщений.

TDSTRIP.EXE

Эта утилита, поставляемая вместе с отладчиком Turbo Debugger for Windows, используется по завершении отладки для исключения из выполняемого файла информации, необходимой только отладчику.

Информация отладчика заносится в выполняемый файл при указании опции **Debug info in EXE** при компиляции с использованием ИСР или опции командной строки компилятора /v в пакетном режиме.

TDUMP.EXE

Эта утилита, также поставляемая вместе с отладчиком, используется для просмотра содержимого файлов различных типов (текстовых, *.EXE*, *.OBJ*, *.DLL*) в различных форматах. Например, при выполнении следующей команды:

```
TDUMP WINAPP.EXE /V
```

можно просмотреть содержимое заголовка Windows-программы.

Ниже приведен фрагмент выводимой утилитой информации вместе с необходимыми комментариями:

```
Заголовок Windows-программы
```

```
New Executable header

Operating system                   Windows
File Load CRC                      000000000h
Program Entry Point   (CS:IP)      0001:0038
Initial Stack Pointer (SS:SP)      0005:0000
Auto Data Segment Index            0005h  (     5. )
Initial Local Heap Size            2000h  (  8192. )
Initial Stack Size                 2000h  (  8192. )
Segment count                      0005h  (     5. )
Module reference count             0004h  (     4. )
Moveable Entry Point Count         005Eh  (    94. )
File alignment unit size           0100h  (   256. )
DOS File Size                      3300h  ( 13056 )
Linker Version                     6.01
.....
```

```
Таблица сегментов
```

```
Segment Table                      offset: 0140h

   Segment Number: 01h
   Segment Type:   CODE            Alloc Size : 003Fh
   Sector Offset:  0003h           File length: 003Fh
   Attributes: Moveable  Preloaded  Relocations

   Segment Number: 02h
   Segment Type:   CODE            Alloc Size : 0EB5h
   Sector Offset:  0004h           File length: 0EB5h
   Attributes: Moveable  Preloaded  Relocations
......
```



```
Таблица ссылок на внешние модули
```

```
Module Reference Table                offset: 016Dh
   Module  1: KERNEL
   Module  2: GDI
   Module  3: USER
   Module  4: KERNEL
   Module  5: USER
   Module  6: KEYBOARD
......
```

Кроме того, утилита TDUMP может быть использована для получения списка функций, реализованных в динамической библиотеке. Ниже показан список таких функций для библиотеки *SYSTEM.DRV*:

```
Name: A20_PROC                 Entry:   20
Name: CREATESYSTEMTIMER        Entry:    2
Name: INQUIRESYSTEM            Entry:    1
Name: SAVE80X87STATE           Entry:    8
Name: KILLSYSTEMTIMER          Entry:    3
Name: GET80X87SAVESIZE         Entry:    7
Name: GETSYSTEMMSECCOUNT       Entry:    6
Name: RESTORE80X87STATE        Entry:    9
Name: ENABLESYSTEMTIMERS       Entry:    4
Name: DISABLESYSTEMTIMERS      Entry:    5
```

Новые типы данных, ключевые слова и директивы компилятора

Для поддержки создания Windows-программ в Turbo Pascal for Windows введен ряд новых типов данных, ключевых слов и директив компилятора.

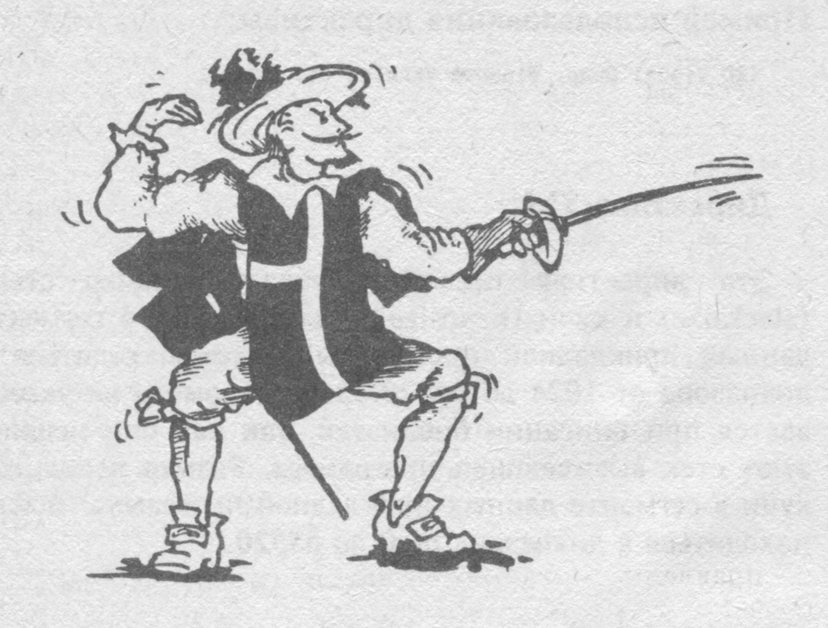
Новые типы данных и операции над ними

В Turbo Pascal for Windows введены следующие новые типы данных:

WordBool — данные типа *Word*, принимающие значения *True* или *False*;

LongBool — данные типа *LongInt*, также принимающие значения *True* или *False*.

В отличие от типа данных *Bool*, принимающего значение *True* только при значении 1, эти типы дан-



ных считаются имеющими значение *True* при любом значении, не равном нулю;

PChar — указатель на тип данных *Char*. Задав директиву расширенного синтаксиса **{$X+}**, пользователь получает возможность выполнять с указателями типа *PChar* ряд операций. Операторы + и - позволяют увеличивать и уменьшать значение смещения указателя внутри сегмента, а оператор - также может быть использован для сравнения двух указателей.

Пример:

```
{Допустимы следующие конструкции:}

Var
CPtr1  :  PChar;
CPtr2  :  PChar;
I      :  Word;
NewPtr :  PChar;
......

NewPtr := CPtr + I;
NewPtr := CPtr - I;

I := CPtr2 - CPtr1;
```

Также для сравнения операндов типа PChar допустимо использование операторов >, <, >= и <=.

Новые ключевые слова

Exports — определяет список процедур и функций, экспортируемых библиотекой.

Пример:

```
library MyDLL;

exports
  function1 index 1;
  function2 index 2;
  .....
  functionN index N;
```

Library — указывает, что модуль является динамической библиотекой, а не прикладной программой (Program) или модулем (Unit).

Export — при использовании этого ключевого слова с описанием процедуры или функции оно указывает, что процедура или функция является экспортируемой. Для экспортируемых процедур и функций компилятор создает специальный пролог и эпилог.

Пример:

```
Function AbortProc(PrinterDC : hDC;Code : Integer) : Bool;
  export;
```

Index — используется для указания числового индекса при описании импортируемой из DLL процедуры или функции.

Пример:

```
Procedure ImportedProc; external 'MYDLL' index $FF;
```

Name — используется для указания имени при описании импортируемой из DLL процедуры или функции.

Пример:

```
Procedure ImportedProc; external 'MYDLL' name 'PROC1';
```

Resident — указывает, что информация об экспортируемой функции должна постоянно находиться в памяти.

Ниже будет приведен пример использования рассмотренных ключевых слов.

Новые директивы компилятора

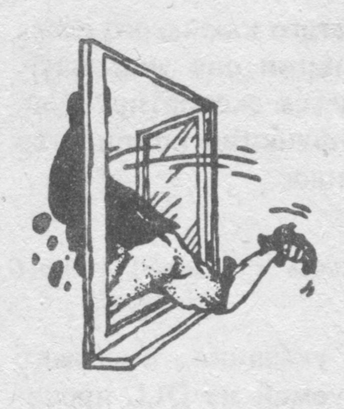
В отличие от компиляторов Borland C++, Turbo C++ for Windows, Microsoft C и ряда других, TPW не использует специальный файл описаний параметров создаваемого модуля (*.DEF*). Такой файл называется module definition file и позволяет указать ряд характеристик создаваемой Windows-программы или библиотеки.

Вместо файла описания модуля в TPW введен ряд новых директив, соответствие которых ключевым словам в DEF-файле приведено в таблице.

DEF-файл	Директива TPW
CODE	$C
DESCRIPTION	$D
EXPORTS	Ключевое слово exports
HEAPSIZE	$M
IMPORTS	Ключевое слово external
STACKSIZE	$M

Директива $C

Использование этой директивы позволяет указать атрибуты сегмента кода. Каждый сегмент кода в при-



кладной программе или библиотеке имеет свой набор атрибутов, определяющих свойства сегмента кода после его загрузки в память. Например, можно указать, что данный сегмент является перемещаемым (moveable). Ядро Windows может перемещать данный сегмент в памяти по мере необходимости. С другой стороны, можно указать, что сегмент является фиксированным (fixed), т.е. его расположение в памяти не должно меняться.

Директива **$C** влияет только на свойства того сегмента (модуля, программы или библиотеки), в котором она расположена. В приводимой ниже таблице перечислены допустимые значения директивы **$C** по группам. Значения внутри каждой группы взаимно исключают друг друга.

MoveAble	Расположение сегмента в памяти может быть изменено
Fixed	Расположение сегмента в памяти не может быть изменено
Preload	Сегмент загружается в память при запуске программы
DemandLoad	Сегмент загружается в память только при необходимости
Permanent	Сегмент всегда остается в памяти после загрузки
Discardable	При необходимости сегмент может быть выгружен из памяти

Принимаемое по умолчанию значение в каждой группе в таблице указано первым. Иными словами, по умолчанию принимаются следующие значения атрибутов:[3]

```
$C MoveAble Preload Permanent
```

Директива $D

Эта директива используется для описания прикладной программы или библиотеки.[4]

Пример использования директивы:

```
{$D Visual Dump. Windows Version (c) 1992}
```

Директива $M

Эта директива позволяет указать размер стека (stacksize) и кучи (heapsize). Размер стека в сегменте данных прикладной программы может находиться в диапазоне от 1024 до 65520. Этот параметр не указывается при описании библиотек, так как они используют стек вызывающей программы. Размер локальной кучи в сегменте данных прикладной программы может находиться в диапазоне от 0 до 65520.

Сумма этих двух величин — размера стека и размера кучи плюс размер всех глобальных переменных не должна превышать 65520 байт, поскольку в Windows-программах под размещение стека, локальной кучи и глобальных переменных отводится только один сегмент данных.

По умолчанию используется директива со следующими параметрами:

```
{$M 8192, 8192}
```

Первое значение указывает размер стека, второе — размер кучи.[5]

Более подробно организация памяти в среде Windows рассматривается в разделе "Управление памятью".

Библиотеки

В состав TPW входит 7 модулей:
- модули **System** и **WinDOS** представляют собой стандартную библиотеку Turbo Pascal и объединены в модуль TPW.TPL;
- модуль **Strings** содержит набор процедур и функций для работы со строками;
- модуль **WinCRT** представляет собой набор подпрограмм, позволяющих быстро преобразовывать простые

[3] Используемые по умолчанию атрибуты более подходят для небольших программ. Директива {$C DemandLoad Discardable} используется в случае, если программа собирается из нескольких модулей. Атрибут **Fixed** нужно указывать лишь при крайней необходимости, поскольку это мешает ядру операционной среды Windows более гибко использовать память.

[4] Текст описания в кавычки не заключается.

[5] Директива $M сохраняет действие только в основном модуле программы или динамической библиотеки. При использовании этой директивы внутри модуля (файла с расширением *.TPU*) она игнорируется.

DOS-программы в Windows-программы (модули Strings и WinCRT также входят в библиотечный модуль TPW.TPL);
- модуль **WinTypes** содержит описание всех типов данных, используемых Windows API;
- модуль **WinProcs** представляет собой интерфейс TPW с функциями Windows API и содержит определения всех документированных функций ядра Windows;
- модуль **WObjects** содержит объектно-ориентированную библиотеку Object Windows.

Рассмотрим каждый из этих модулей более подробно.[6]

Модуль System

Модуль **System** содержит подпрограммы, управляющие файловым вводом/выводом, распределением памяти и работой с математическим сопроцессором. Модуль System автоматически подключается компилятором к любой программе (поэтому он должен всегда находиться в составе стандартной библиотеки TPW.TPL). Отметим, что в модуле System для TPW имеется ряд уникальных для Windows переменных:

- **CmdShow**. Параметр, указывающий на режим начального отображения окна.
- **hIstance**. Указатель экземпляра программы.
- **hPrevInst**. Указатель предыдущего экземпляра программы.

Эти три переменные передаются основной функции любой Windows-программы (WinMain) при ее вызове ядром Windows.

В комплект поставки включен файл *SYSTEM.INT*, содержащий описание интерфейсной части этого модуля.

Модуль WinDOS

Модуль **WinDOS** содержит ряд подпрограмм для работы с файлами и подпрограммы доступа к средствам операционной системы. Этот модуль необходим при работе с регистрами центрального процессора (тип данных *TRegisters*), файлами на уровне DOS (тип данных *TFileRecord*), системной датой (тип данных *TDateTime*) и векторами прерываний (процедуры **GetIntVect**, **Intr**, **MsDOS** и **SetIntVect**). Модуль содержит функции определения размера диска (**DiskSize**) и свободного места на диске (**DiskFree**), функции работы со средой командного процессора, параметрами командной строки (**GetArgCount**, **GetArgStr** и **GetEnvVar**) и подкаталогами (**CreateDir**, **RemoveDir** и **SetCurDir**). По своему функциональному назначению этот модуль соответствует модулю DOS, поставляемому с компилятором Turbo Pascal 6.0. Имена переменных и функций изменены в соответствии с принятыми в системе Windows и библиотеке Object Windows соглашениями. Имена всех типов данных начинаются с буквы *T* (*Type*). Таким образом, в TPW тип данных *TRegisters* соответствует типу данных Registers в TP6.0. Более подробно отличия в именовании типов данных и названий процедур рассматриваются в главе, посвященной файловому вводу/выводу.

[6] Чтобы получить исходные тексты перечисленных выше модулей, достаточно приобрести **Turbo Pascal for Windows Runtime Library**, которая распространяется среди всех зарегистрированных пользователей **Turbo Pascal for Windows**. Поставляются исходные тексты библиотеки Object Windows и модулей System, WinDOS, WinCRT, WinTypes, WinProcs и Strings.

воду. В комплект поставки включен файл *WINDOS.INT*, содержащий описание интерфейсной части модуля.

Модуль Strings

Модуль **Strings** содержит подпрограммы для работы со строками, заканчивающимися нулевым байтом (ASCIIZ-строки), которые используются в качестве параметров ряда функций **Windows API**.

Обычный для языка Pascal тип данных *string* представляет собой массив символов, нулевой элемент которого содержит размер строки. ASCIIZ-строки содержат последовательность символов и заканчиваются нулевым байтом (#0). Тогда как стандартный тип данных *string* ограничен размером в 255 байт, ASCIIZ-строки могут занимать до 64 Кбайт. Для преобразования стандартных строк в ASCIIZ-строки используется функция **StrPCopy**, а для обратного преобразования — функция **StrPas**.

Стандартные процедуры **Read**, **Readln**, **Write** и **Writeln** поддерживают как обычные, так и ASCIIZ-строки. В модуль включены функции для копирования и объединения строк, для определения длины строки, сравнения и преобразования строк.

Файл *STRINGS.INT* содержит описание интерфейсной части данного модуля, включая откомментированные прототипы всех процедур и функций, вошедших в данный модуль.

Модуль WinCRT

Модуль **WinCRT** может быть использован для преобразования DOS-программ, осуществляющих ввод/вывод с помощью функций **Read/Write**, **Readln/Writeln**, и создания небольших Windows-программ.

Все, что необходимо сделать для преобразования DOS-программы, отвечающей указанным выше требованиям, в Windows-программу, — это включить в начале программы следующую строку:

```
uses WinCrt;
```

В результате ввод/вывод в стандартные файлы *Input* и *Output*, определенные в модуле **System**, будет перенаправлен в специальное окно, которое эмулирует экран в текстовом режиме. При запуске программы создается окно, в заголовке которого указывается имя выполняемого файла. По достижении конца программы заголовок меняется на *Inactive nnnnn*, где *nnnnn* — имя выполняемого файла. Завершение работы с программой происходит при закрытии окна.

Окно создается при вызове функции **InitWinCrt**, а работа с ним завершается с помощью функции **DoneWinCrt**.

Простейшая программа, использующая модуль **WinCRT**, будет выглядеть следующим образом:

```
{$X+}
uses WinCrt;
Begin
    GotoXY(ScreenSize.X div 2, ScreenSize.Y div 2);
    Write('Hello from WinCRT !');
    ReadKey;
End.
```

Примечания:

Директива **{$X+}** указывает на необходимость использования расширенного синтаксиса — результат, возвращаемый функцией, может игнорироваться (функция используется как процедура), что показано в предпоследней строке при вызове функции **ReadKey**.

Параметры *ScreenSize.X* и *ScreenSize.Y* задают размер экрана по горизонтали и вертикали соответственно. Чтобы задать необходимый размер экрана, можно использовать следующие команды:

```
ScreenSize.X := 80;
ScreenSize.Y := 25;
```

При установке размера экрана необходимо помнить, что величина *ScreenSize.X * ScreenSize.Y* не должна превышать значения 65520.

Для изменения заголовка окна, который описывается в массиве символов WindowTitle, можно использовать следующий вызов:

```
StrCopy(WindowTitle,'WinCrt demo');
```

Процедура **StrCopy** находится в модуле **Strings**.

Чтобы заменить заголовок неактивного окна, можно указать новое значение переменной *InactiveTitle* (значение по умолчанию *"(Inactive %s)"*):

```
StrCopy(InactiveTitle,'WinCrt demo');
```

При использовании модуля **WinCRT** возможно использование стандартных процедур ввода/вывода **Read**, **Readln**, **Write** и **Writeln**. В модуле **WinCRT** также реализованы функции **ReadKey** и **ReadBuf**.

Другие переменные модуля **WinCRT** содержат следующие значения:

WindowOrg — тип данных *TPoint*. Содержит начальные координаты окна. Значения по умолчанию: *X := cw_UseDefault, Y := cw_UseDefault*.

WindowSize — тип данных *TPoint*. Содержит начальный размер окна. Значения по умолчанию: *X := cw_UseDefault, Y := cw_UseDefault*.

ScreenSize — тип данных *TPoint*. Содержит размер окна по горизонтали и вертикали в символах. Значения по умолчанию: *X := 80, Y := 25*.

Cursor — тип данных *TPoint*. Содержит текущую позицию курсора. Коодината левого верхнего угла считается точкой (0,0). Значение этой константы изменять нельзя.

Origin — тип данных *TPoint*. Содержит координату левой верхней точки экрана (0,0). Значение этой константы изменять нельзя.

AutoTracking — тип данных *Boolean*. Управляет автоматической прокруткой содержимого экрана.

Некоторые советы по использованию модуля WinCRT

- Для вывода одного символа *Ch* более эффективно использовать процедуру **WriteCh(Ch)**, которая работает быстрее **Write(Ch)**.
- Для ввода символа *Ch* более эффективным является использование функции **Ch := ReadKey**.
- Две процедуры **GotoXY** и **CursorXY** используют различные точки отсчета: **GotoXY** считает левую верхнюю позицию на экране точкой с координатами (1,1), тогда как **CursorXY** начинает отсчет от точки с координатами (0,0). Использование процедуры **CursorXY** является более эффективным.
- Функции **WhereX** и **WhereY** используются совместно с процедурой **GotoXY** и производят отсчет от точки с координатами X=1, Y=1. При использовании процедуры **CursorXY** необходимо использовать поля переменной *Cursor: Cursor.X* и *Cursor.Y*.
- При установке значения *AutoTracking := True* содержимое экрана будет автоматически прокручиваться при выводе в невидимую область. При значении *AutoTracking := False* для просмотра информации необходимо использовать полосы прокрутки или процедуру **ScrollTo**.[7]

В приводимой ниже программе **WinMetr** показано, как, используя модуль **WinCRT**, создать небольшую утилиту, которая отображает следующую информацию:

- тип процессора;
- текущий режим работы среды Windows;
- тип клавиатуры и число функциональных клавиш;
- текущую версию Windows;
- различные системные метрики.[8]

```
{————————————————————————————————————————————————————————
      Программа WINMetrics
      Программа для определения различных системных параметров
         Компилятор      : Turbo Pascal for Windows.
         Версия Windows : Windows 3.0
      Используется модуль WinCRT
————————————————————————————————————————————————————————}
Program WinMetrics;
uses WinCRT,WinTypes,WinProcs,Strings;
Var
       Flags   : LongInt;   {Флаги состояния Windows}
       Ch      : Char;
{----   Показать тип процессора (wf_CPUXXX) ----}
Procedure ShowCPU;
Var
       CPU : PChar;
Begin
{}          if Flags and wf_CPU086 <> 0 then CPU := '8086'
       else if Flags and wf_CPU186 <> 0 then CPU := '80186'
       else if Flags and wf_CPU286 <> 0 then CPU := '80286'
       else if Flags and wf_CPU386 <> 0 then CPU := '80386'
       else if Flags and wf_CPU486 <> 0 then CPU := '80486'
       else CPU := 'Unknown';

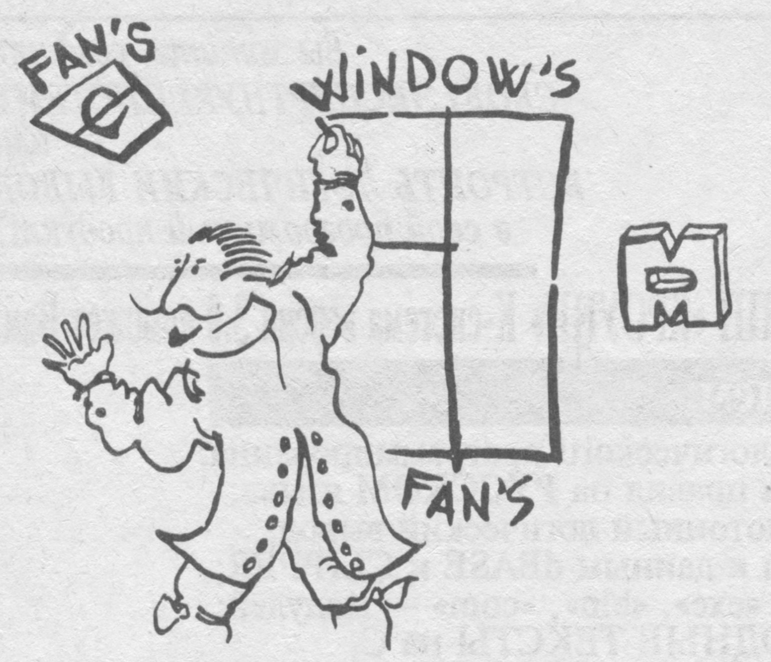
     Writeln;
     GotoXY(20,WhereY);Write('CPU Type: ');
       GotoXY(50,WhereY);Writeln(CPU);
End;

{---- Показать режим работы Windows (wf_XXX) ----}
Procedure ShowMode;
Var
       Mode : PChar;
Begin
       if Flags and wf_Enhanced <> 0 then Mode := 'Enhanced'
       else if Flags and wf_Standard <> 0 then Mode := 'Standard'
       else Mode := 'Standard';
          GotoXY(20,WhereY);Write('Operating mode:');
          GotoXY(50,WhereY);Writeln(Mode);
End;
{---- Показать тип клавиатуры и число функциональных клавиш ----
}
Procedure ShowKBD;
Var
       KBD : PChar;
Begin
          GotoXY(20,WhereY);
             Write('Keyboard type');
        case GetKeyboardType(0) of
            1 : Kbd := 'IBM PC/XT 83 Key';
            2 : Kbd := 'Olivetti M24 102 Key';
            3 : Kbd := 'IBM AT 84 Key';
            4 : Kbd := 'IBM Enhanced 101/102 Key';
            5 : Kbd := 'Nokia 1050';
            6 : Kbd := 'Nokia 9140';
            else Kbd := 'Unknown';
       end;

          GotoXY(50,WhereY);Writeln(Kbd);
          GotoXY(20,WhereY);Write('Number of functional keys: ');
```

[7] Модуль **WinCRT** поставляется в исходном виде (файл *WINCRT.PAS*).

[8] Режимы работы операционной среды Windows рассматриваются в главе "Управление памятью", системные метрики более подробно описываются в главе, посвященной созданию окон.




```
     GotoXY(50,WhereY);Writeln(GetKeyboardType(2));
End;
{---- Показать версию Windows ----}
procedure ShowWinVer;
Var
    Version : Integer;  {Hold Windows Version}
Begin
     Version := GetVersion;
   GotoXY(20,WhereY);   Write('Windows version :');
   GotoXY(50,WhereY);
   Write(LoWord(Version),'.',HiWord(Version));
End;

{Показать различные метрики (sm_XXX)}
Procedure ShowSysMetrics;
Procedure ShowSM(sm_Name : PChar;sm_Index : Integer);

Begin
      GotoXY(15,WhereY);Write(sm_Name);
     GotoXY(60,WhereY);Writeln(GetSystemMetrics(sm_Index));
End;

{Булевские значения отображаются как Yes (True) или No (False)}
procedure ShowBoolean(sm_Name : PChar; sm_Index : Integer);
Var
    Bool : PChar;
Begin
     GotoXY(15,WhereY);Write(sm_Name);
   if GetSystemMetrics(sm_Index) = 0 then
        Bool := 'Yes'
   else
        Bool := 'No';
     GotoXY(60,WhereY);Writeln(Bool);
End;

Begin**        Writeln('{sm_XXX}');;
    ShowSM('Screen Width (sm_CXScreen):', sm_CXScreen);
  ShowSM('Screen Height (sm_CYScreen):', sm_CYScreen);
  ShowSM('Max X-size (sm_CXFullScreen):',sm_CXFullScreen);
  ShowSM('Max Y-size (sm_CYFullScreen):',sm_CYFullScreen);
  ShowSM('Min X-size (sm_CXMin):',sm_CXMin);
  ShowSM('Min Y-size (sm_CYMin):',sm_CYMin);
  ShowSM('Cursor width (sm_CXCursor):',sm_CXCursor);
  ShowSM('Cursor height (sm_CYCursor):',sm_CYCursor);
  ShowSM('Icon width (sm_CXIcon):', sm_CXIcon);
  ShowSM('Icon height (sm_CYIcon):', sm_CYIcon);
  ShowBoolean('Mouse installed (sm_MousePresent):',
             sm_MousePresent);
  ShowBoolean('Mouse buttons swapped (sm_SwapButtons):',
              sm_SwapButton);
    Writeln('{wf_XXX}');
  ShowBoolean('Protected mode (wf_PMode):',wf_PMode);
  ShowBoolean('EMS large frame (wf_LargeFrame):',wf_LargeFrame);
  ShowBoolean('EMS small frame (wf_SmallFrame):',wf_SmallFrame);

End;
```

```
Begin
    {Set new Window header}
{Window Title}
   StrCopy(WindowTitle,'WinMetrics');
     CmdShow := sw_Maximize; {Show Window Maximized}
{Open Window}**       InitWinCRT;
     GotoXY(30,WhereY);Writeln('WINDOWS METRICS');
     GotoXY(30,WhereY);Writeln('---------------');
   ShowWinVer;
   Flags := GetWinFlags;
   ShowCPU;
   ShowMode;
   ShowKBD;
   ShowSysMetrics;
   Ch := ReadKey;DoneWinCRT
End.
```

Модуль WinTypes

Модуль WinTypes содержит описание всех типов данных, используемых **Windows API**. В нем описаны константы, структуры данных, типы данных и сообщения. На дискетах, входящих в комплект поставки, находится интерфейсная часть этого модуля (файл *WINTYPES.INT*). Этот файл может быть использован в качестве справочного руководства.

Модуль WinProcs

Этот модуль служит интерфейсной частью между **TPW** и функциями **Windows API**. В нем описаны прототипы всех документированных функций интерфейса Windows. Этот модуль поставляется в исходном виде (файл *WINPROCS.PAS*). Вместе модули WinTypes и WinProcs являются эквивалентом файла *WINDOWS.H*, поставляемого в составе Microsoft Windows Software Development Kit (SDK).



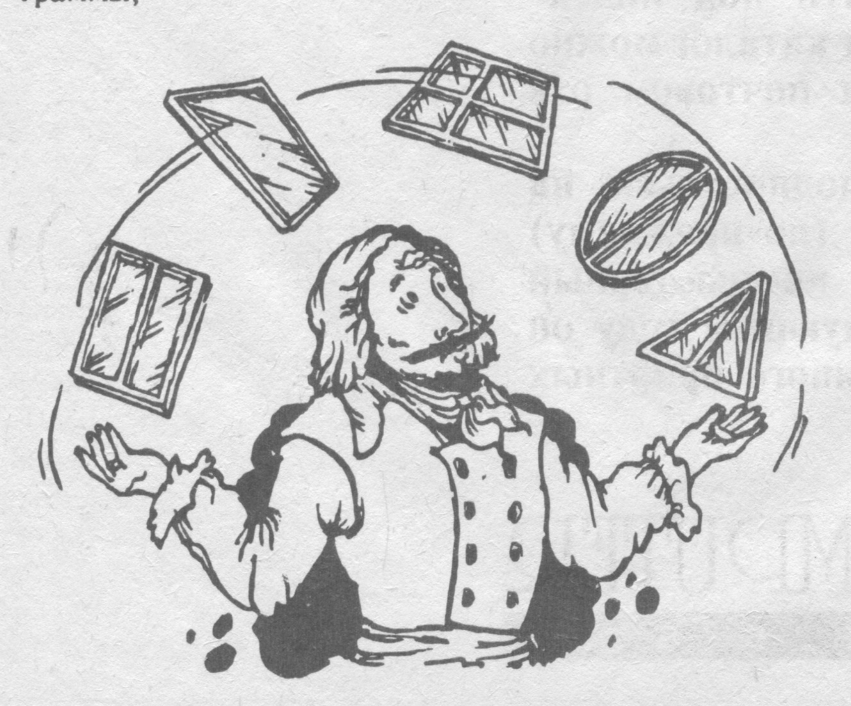
Модуль WObjects

Модуль WObjects содержит объектно-ориентированную библиотеку для создания Windows-программ Object Windows. Более подробно эта библиотека рассматривается на протяжении всей книги, здесь же мы остановимся на основах организации этой библиотеки. (Файлы, содержащие интерфейсные части перечисленных выше модулей, находятся в подкаталоге *C:\TPW\DOC*, создаваемом при установке компилятора программой INSTALL.)

Библиотека Object Windows

Object Windows — это объектно-ориентированная библиотека, позволяющая существенно облегчить создание Windows-программ. Отметим, что эта библиотека облегчает лишь часть работы по написанию прикладной программы — созданию ее интерфейсной части (окон, панелей диалога и элементов управления) и позволяет упростить обработку и диспетчеризацию сообщений. Вся остальная "деятельность" программы основывается на непосредственных вызовах функций Windows API.

Можно выделить четыре аспекта, которые учитывались при разработке Object Windows:
- упрощение процесса обработки сообщений;
- сокращение времени по созданию основных компонентов Windows-программы;



- поддержка стандартного пользовательского интерфейса;
- полный доступ к функциям Windows API.

Библиотеку Object Windows можно рассматривать как оболочку прикладной программы, не выполняющую никаких действий. Библиотека содержит процедуры инициализации, обработки сообщений и не ограничивает возможность расширения свойств уже существующих объектов.

Библиотека Object Windows состоит из более чем 20 типов объектов, организованных в виде иерархического дерева. Объекты, входящие в состав библиотеки Object Windows, и процесс создания объектно-ориентированных Windows-программ рассматриваются в соответствующих разделах данной книги.

Подход, реализованный в библиотеке Object Windows, предложен фирмой Borland в качестве стандарта

на объектно-ориентированную оболочку для создания Windows-программ и находится на рассмотрении специального комитета Object Management Group.

Отметим, что оригинальным разработчиком библиотеки Object Windows является фирма Whitewater Group, и первоначальная версия библиотеки была реализована для объектно-ориентированного языка Actor, созданного этой фирмой. Фирма Whitewater Group выпустила также библиотеку Object Graphics, расширяющую стандартный набор графических функций, предоставляемых ядром Windows.

Документация

Поставляемая в пакете документация состоит из 6 книг.

1. *Руководство пользователя (User's Guide)* описывает процедуру установки и настройки компилятора. Приводятся описание языка Pascal, введение в объектно-ориентированное программирование и справочник по интегрированной среде разработчика.

2. *Руководство программиста (Programmer's Guide)* содержит описание стандартной библиотеки, интерфейса с языком ассемблера и другую полезную информацию. Приводятся описание всех опций компилятора и объяснение сообщений об ошибках. Кроме того, имеется список (в алфавитном порядке) всех процедур и функций, входящих в состав стандартной библиотеки *TPW.TPL*.

3. *Руководство по программированию в среде Windows (Windows Programming Guide)* посвящено описанию технологии создания Windows-программ с использованием библиотеки Object Windows.

4. *Справочное руководство Windows (Windows Reference Guide)* содержит описание всех констант, стилей, сообщений и документированных функций Windows API, а также полный справочник по объектам и методам объектно-ориентированной библиотеки Object Windows.

5. *Редактор ресурсов (Whitewater Resource Toolkit)* описывает процедуры создания и редактирования различных типов ресурсов.

6. *Руководство пользователя Turbo Debugger for Windows (Turbo Debugger for Windows User's Guide)* содержит информацию об отладке Windows-программ.

Помимо перечисленной выше документации на диске в подкаталоге *C:\TPW\DOC* находится ряд файлов (*HELPME!.TDW*, *MANUAL.TDW*, *MANUAL.TPW*), содержащих ответы на наиболее часто задаваемые вопросы и изменения в документации. Информация о всех файлах, входящих в комплект поставки, находится в файле *READ.ME*, располагаемом в каталоге *C:\TPW*.

Д.Рогаткин, А.Федоров



QuickTime открывает новый этап

MPC, Multimedia Windows и другие шаги Microsoft в области мультимедиа являются попыткой — возможно, несколько запоздалой — привнести некоторый порядок в бурное и беспорядочное внедрение элементов мультимедиа на персональных компьютерах линии IBM. Давайте посмотрим, как решает подобные проблемы фирма Apple — чтобы в сопоставлении лучше понять подходы фирм, уровни постановки задач и пути их решения.

HyperCard была первой, очень продуманной, удобной и, видимо, наиболее распространенной средой для работы с мульти- и гипермедиа в годы зарождения технологии. Теперь средства мультимедиа становятся органичной частью операционной системы. Apple разработала и выпустила на рынок QuickTime — расширение MacOS System 7, новую архитектуру программного обеспечения для работы с динамическими данными: звуком, видео, анимацией. Еще прошлым летом, при знакомстве с первыми предварительными описаниями будущего продукта [1-3], становилось ясно, что на подходе система, открывающая новый этап в развитии мультимедиа. Обозреватели и аналитики ждали появления в мире мультимедиа так называемой "killer application" — прикладной программы, которая вызовет широкое внедрение мультимедиа, сыграет роль приманки для потребителей, подобную той, которую сыграл в судьбе IBM PC Lotus 1-2-3. Вместо этого появилась QuickTime, которую можно скорее определить как "killer environment" — среду, фундамент, который обеспечит естественное и повсеместное внедрение мультимедиа.

QuickTime состоит из четырех основных разделов: Системные программы, Форматы файлов, Компрессоры Apple и Стандарты интерфейса пользователя (Human Interface Standards); архитектура открыта, расширяема, предлагает принципы обмена данными мультимедиа между различными стандартами и форматами.

Системные программы, в свою очередь, делятся на три блока. Первый, Movie Toolbox ("киноинструмент") — это набор инструментальных средств и служб, позволяющих прикладным программам обеспечить создание, редактирование и воспроизведение динамических типов данных. Второй, Image Compression Manager (менеджер сжатия изображений), включает независимые от устройств и алгоритмов службы, с их помощью прикладные программы могут без модификации использовать различные методы сжатия/расширения — DVI, MPEG, JPEG, собственные Apple и др. Третий блок, Component Manager, позволяет внешним ресурсам (платам расширения, дигитайзерам, видеомагнитофонам, лазерным плеерам, софтверным драйверам) с помощью специальных интерфейсных протоколов регистрироваться в системе, сообщать ей свои возможности в процессе выполнения программы — таким образом, прикладные программы ограждены и от особенностей применяемых внешних устройств и плат мультимедиа.

Форматы файлов — это набор стандартных описаний для динамических данных, позволяющий осуществлять перенос данных между различными программами, документами, файлами.

Вводится новый тип данных, "movie" ("фильм"). Movie состоит из произвольного числа параллельных "дорожек" (track), содержащих, как правило, указатели на элементы презентации (а не сами элементы) — образы, звуки, анимационные данные и т.п. Это позволяет задавать сложные и длительные презентации в небольших movie-файлах. Элементы презентации обычно хранятся в сжатом виде; сжатие и распаковка осуществляются компрессорами Apple. Фильмы могут создаваться в диалоге с помощью Movie Editor (редактора фильмов), либо генерироваться ав-

Продолжение Начало в №7'92 (Мультимедиа-91: достижения, тенденции, рынок).

томатически прикладной программой. Введение новых типов данных, новых элементов легко обеспечивается добавлением дополнительных параллельных дорожек. Структура дорожек делает более надежной синхронизацию событий, синхронность видеоряда и звукового сопровождения. Фильмы QuickTime "настраиваемы" на воспроизводящую среду, проигрываются с максимально доступным на данной установке качеством. Системные службы выбирают требуемые модули декомпрессии для элементов данных, при необходимости "огрубляют" цвета изображения для воспроизведения с помощью имеющейся графической платы, пропускают некоторые кадры видеопоследовательности, чтобы не потерять синхронизацию со звуком или анимацией, если имеющееся на компьютере оборудование не успевает распаковывать видео с требуемой скоростью. Формат Apple для неподвижных изображений, PICT, расширен и допускает сжатие данных; теперь любая программа, работающая с PICT-форматом, может хранить данные в сжатом виде — службы QuickTime распакуют их.

Компрессоры Apple — новые программы сжатия/расширения, поставляемые с QuickTime, включающие фото-, видео-, анимационные и графические "кодеки" (codec, compressor-decompressor)[1]. Для фото- и видеоданных обеспечивается сжатие до 25:1. Фотокодек основан на JPEG, анимационный — на алгоритме RLE; видеокодек реализует собственный алгоритм Apple; текущая его версия может проигрывать на Mac-si 15 кадров/с с разрешением 160x120; более быстрые модели обеспечат лучшее качество. Готовятся новые программные кодеки, реализующие различные алгоритмы сжатия. При наличии аппаратных кодеков — плат JPEG, DVI и др. — Compression Manager автоматически выбирает их.

Стандарты интерфейса должны облегчить создание диалоговых приложений, управлять проигрыванием фильмов, захватом и сжатием изображений. Сюда относятся также средства Preview — быстрого просмотра. Они обеспечивают хранение в сжатом виде не только самих изображений, но и их уменьшенных, с почтовую марку, копий; эти копии используются в качестве пиктограмм — они выводятся на экран вместо символьных оглавлений, позволяя просматривать банки изображений со скоростью просмотра символьных оглавлений. Стандарты интерфейса призваны, кроме того, обеспечить некоторое единообразие взаимодействия с пользователем для разных программ от разных производителей. Можно создавать и собственные интерфейсы — все детали "внутреннего устройства" QuickTime максимально открыты. Видимо, Apple усвоила уроки прошлого, когда излишняя закрытость сильно помешала компьютерам Mac в борьбе с IBM PC[2].

Подход Apple, как всегда, отличают комплексная стратегия, забота о программистах и пользователях, наилучшее, пожалуй, на сегодня понимание проблем технологии мультимедиа. QuickTime должна не только значительно облегчить жизнь разработчикам продуктов мультимедиа, но и, что более важно, сделать элементы мультимедиа естественными для любой программы. От разработчика не требуется, как в случае с Windows, специальных усилий при создании и особенно при модификации программы для встраивания ее в жесткие рамки среды — QuickTime предстает просто как расширение, дополнительный сервис операционной системы. Благодаря этому можно ожидать "обвального" внедрения QuickTime. Уже в начале года, к моменту официального представления архитектуры на MacWorld Expo, свыше 160 продуктов были зарегистрированы фирмой Apple как QuickTime-совместимые; подавляющее большинство фирм собиралось в ближайшее время обеспечить работу с QuickTime (вспомним, сколько времени требуется для создания Windows-версий существующих программ для IBM PC).

Естественно, сразу приняли QuickTime разработчики анимационных и графических программ, настольных видеостудий. На основе средств QuickTime усовершенствуется интерфейс, встраивается работа с новыми типами данных в базы данных и информационные системы. В HyperCard вводятся новые XCMD для поддержки фильмов QuickTime. Даже текстовые процессоры (WordPerfect, Word) уже используют фильмы QuickTime.

Разработчики оборудования, в частности, крупнейшие производители видеоплат для Mac — SuperMac, Videologic, Radius, RasterOps и др. — также срочно готовят драйверы QuickTime для своих устройств. QuickTime уже начинает применяться в системах видеоконференций и электронной почты: фирма Compression Lab демонстрировала систему MacVideoPhone1 (1460 долл.), которая позволяет передавать видеоизображение по стандартным телефонным линиям, а при желании — и запоминать его в виде фильма QuickTime.

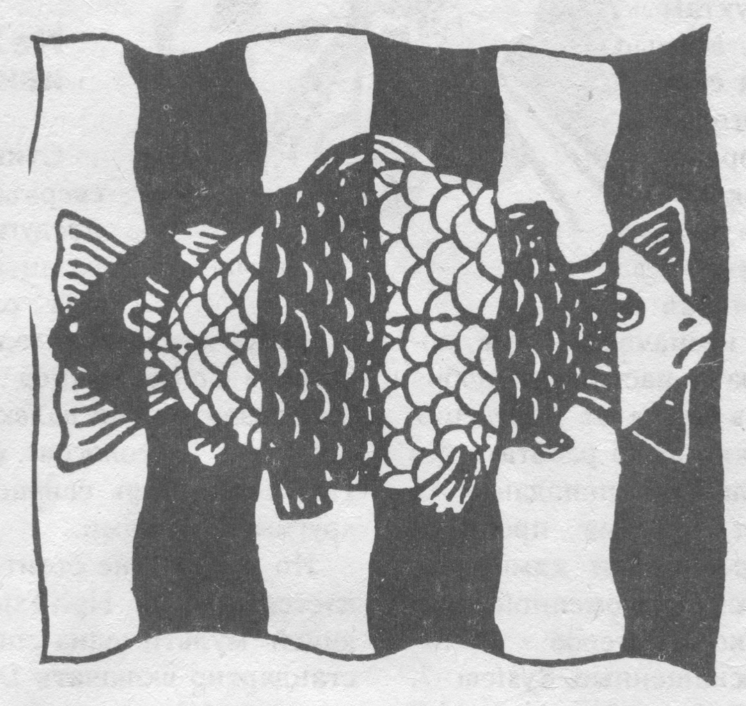
QuickTime обладает большей, по сравнению с Windows, "адаптивностью" к новым техническим сред-

---

[1] В некоторых источниках "codec" употребляется и как сокращение от "encoder/decoder" (шифратор/дешифратор).

[2] Хотя возможен и несколько иной угол зрения на эту проблему. Открытость архитектуры IBM PC, безусловно, стала одной из основных причин широчайшего распространения компьютеров этой линии, завоевания ими порядка 80% мирового рынка; Apple же, защитив свою архитектуру, вынуждена была вести борьбу в одиночку, контролируя лишь около 15-18% рынка. Однако самой фирме IBM открытость, так помогавшая на первом этапе, больших преимуществ в конце концов не дала: утеряв контроль над рынком, теснимая безудержным распространением более дешевых и быстрых клонов, фирма вынуждена была перейти на закрытую архитектуру MCA, оставив прежнее поле боя, вместе с архитектурой ISA, многочисленным фирмам-эпигонам (эти-то фирмы, кстати, и образуют теперь консорциум MPC — во главе с покинувшей IBM Microsoft); доля самой IBM упала с более чем 40 до 23% [4]. А Apple продолжает стабильно удерживать свою долю рынка, поддерживая и развивая линию Mac.

ствам, технологиям во многом благодаря свойствам всей окружающей операционной среды. Мы рассматривали ранее в этом и предыдущем обзорах различные технические и технологические достижения, которые позволяют улучшить работу с компьютером, с новыми типами данных; эти достижения должны, однако, както внедряться, объединяться, уживаться с ранее созданными продуктами, с системами, далеко не всегда позволяющими это легко осуществить. QuickTime же, вместе со своим окружением, являет собой "опережающую" идеологию, новую принимающую среду, которая готова — и предназначена — внедрить, впитать в себя и эти, и многие будущие достижения.



Это, впрочем, еще одно проявление общего различия в основополагающих подходах — "от производителя" и "от пользователя". Вспомним, например, историю развития графики на IBM и Mac. Пользователи и программисты IBM-совместимых компьютеров вынуждены были следовать мучительному зигзагообразному пути разработчиков графических плат, меняя программы, дизайн и даже сами мониторы. Сначала был CGA — с чрезвычайно неудобным раздельным хранением четных и нечетных строк изображения — из-за того только, что разработчики решили сэкономить (применили дешевый чип 6845, а чтобы он все-таки успевал работать, придумали изящное инженерное решение с разделением банков памяти; помоему, эта "экономия" многократно была оплачена программистскими слезами). Затем последовали EGA с цифровым монитором, странный гибрид MCGA, VGA — уже с аналоговым выходом, SVGA, теперь вот XGA... Натурального 24-битного цвета по-прежнему нет, разные контроллеры и режимы требуют разного программирования и представления данных, попытки работать с высоким разрешением приводят к появлению нечитаемых микробукв, так как системные средства "не видят" переключения режима, и т.д. Apple же уже несколько лет тому назад ввела единый 32-битный формат графических данных QuickDraw, благодаря чему пользователь заботится только о "содержательной" стороне, а средства операционной системы самостоятельно настраиваются на имеющиеся возможности компьютера.

В новые, недавно объявленные планы Apple, входит "модуляризация" Mac OS. Такие расширения системы, как OCE (Open Collaborative Environment, открытая среда взаимодействия — новая стратегия Apple в области сетевого взаимодействия), QuickTime, AppleScript, Imaging model (дальнейшее развитие QuickDraw), будут поставляться отдельно, по выбору пользователя, формируя ОС "под заказчика"; разрабатываются модули речевого и рукописного ввода и другие, пока широко не объявленные. Apple также объявила о разработке Windows-версий этих модулей (в первую очередь, AppleScript, QuickTime и OCE), чем вызвала некоторое замешательство в рядах фирм, ориентирующихся на Macintosh: они усмотрели в этом сдачу позиций и чуть ли не переориентацию Apple на Windows. Руководству Apple пришлось даже выступить с серией заявлений и интервью, подтверждающих, что Macintosh будет оставаться в центре интересов фирмы по крайней мере в течение ближайших 5 лет [7-9].

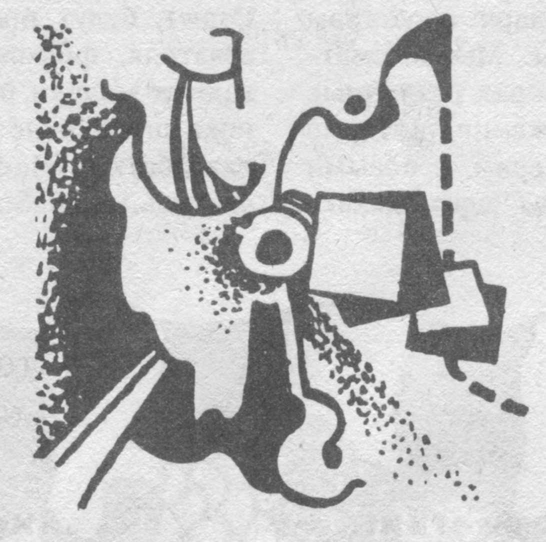


Новый подход Apple к развитию системного софтвера представляется мне проявлением корреляции долговременной политики фирмы с ожидаемыми плодами деятельности Taligent и других совместных с IBM фирм. Чтобы плоды эти не обломили сук, на котором сама же Apple сидит, предусмотрительные хозяева заранее начали подготовку. Чем эти меры предосторожности в конце концов обернутся — надежными подпорками или подстеленной соломкой — покажет время. Согласно сообщениям руководства Apple, по крайней мере два-три года должно пройти, прежде чем работы Taligent принесут практические результаты. Все ждут их с нетерпением, как ждали бы, наверное, выхода совместной картины Бондарчука и Феллини, гадая о том, на что же ЭТО может быть похоже. Pink и PowerPC стали очень модной темой для обсуждения. А Mac-ориентированные фирмы испытывают не только интерес, но и определенное беспокойство: они помнят неожиданный и резкий переход Apple от линии AppleII к линии Mac в 1984 году. Тем не менее, наблюдая за изменениями в поведении Apple и IBM в номенклатуре и идеологии вновь выпускаемых продуктов, можно попробовать кое-что предугадать.

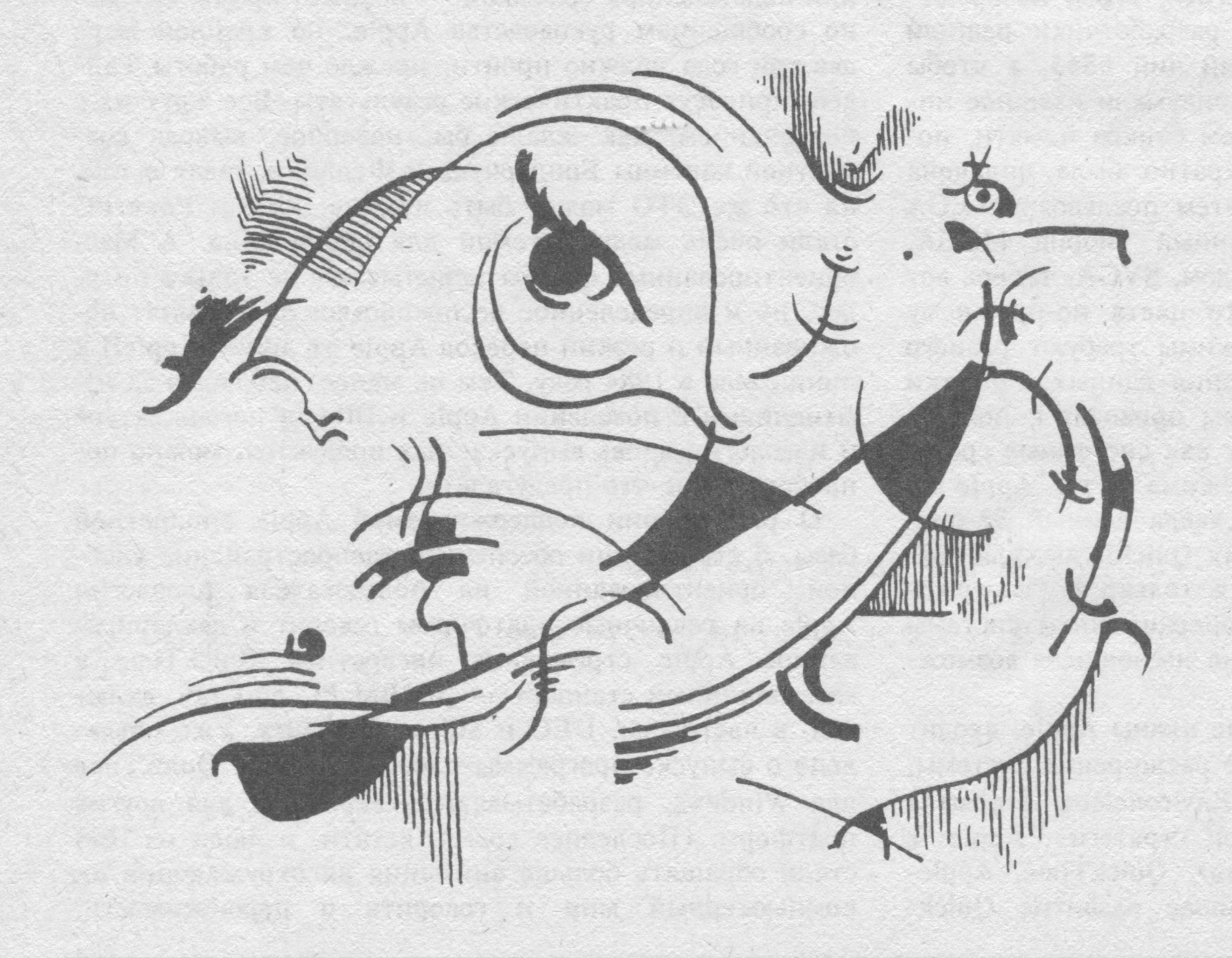
О расширении поддерживаемой Apple аппаратной базы, о стремлении обеспечить распространение удобной, ориентированной на пользователя идеологии Apple на различные платформы говорит и декларированное Apple стремление превратить QuickTime в межмашинный стандарт — от IBM PC до Cray, включая, в частности, DEC и Silicon Graphics. Уже объявлено о выпуске программы-плеера фильмов QuickTime для Windows; разрабатываются версии и для других платформ. (Последнее время, кстати, и люди из IBM стали обращать больше внимания на окружающий их компьютерный мир и говорить о переносимости,

"межплатформенной" совместимости программного обеспечения, о поддержке усилий IMA в этом направлении.)

Руководитель Apple John Sculley недавно сказал, что Apple его времени запомнят не по компьютерам Macintosh, не по каким-то еще продуктам фирмы, а — по QuickTime [10]. Мне кажется, QuickTime являет собой счастливый пример системы, которая создавалась людьми, очень хорошо представлявшими себе и поставленную задачу, и среду, в которую продукт должен вписаться, и будущего потребителя; системы, которая настолько соответствует своему назначению, так дополняет возможности компьютера и настолько удобна для пользователя, что через день после ее появления уже трудно представить, как можно было работать без нее (известный всем пример подобного попадания "в яблочко" — Norton Commander). Такие продукты практически не нуждаются в рекламной кампании, они как бы спонтанно становятся непременной принадлежностью практически всех компьютеров.



Компьютеры фирмы Apple, оснащенные System 7, включающей архитектуру взаимодействия IAC (InterApplication Communication Architecture), QuickTime, OCE и др., обладающие множеством различных авторских и инструментальных сред и систем, сегодня, пожалуй, являются наиболее универсальным и комфортным инструментом. Во многом Apple следует направлению, которое ее основатель Стив Джобс указал развитием системы NeXT. Сегодняшние подходы Apple по существу являются воплощением идей Джобса об "интерперсональном компьютинге".



NeXT как путеводная звезда

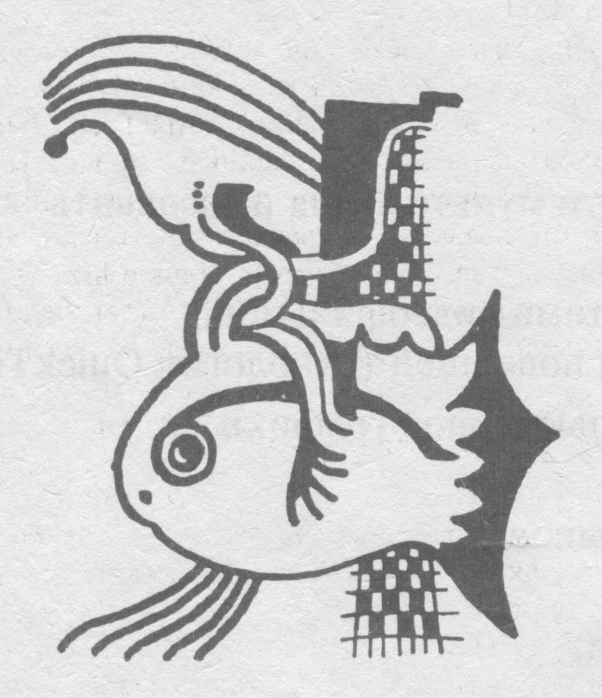
Стив Джобс, наверное, обладает сверхъестественной способностью предугадывать (или задавать?) путь развития рынка и технологии. В третий раз он сделал резкий шаг в сторону, начал прокладывать собственную дорогу — и опять в конце концов оказался на главном направлении развития, но далеко впереди остальных. Интегрированная идеология системы NeXT, созданной в 1988 году, лишь сейчас воспроизводится Apple, Sun и другими фирмами.

Но и NeXT не стоит на месте: в середине года ожидается выпуск NeXTstep 3.0, еще дальше продвигающей мультимедиа-способности системы. Она будет стандартно включать Display PostScript Level 2, систему построения и анимации трехмерных образов Pixar RenderMan 3D (работающую как в "интерактивном", так и "фотореалистическом" режимах), встроенные протоколы NetWare и AppleShare (т.е. NeXT без всяких дополнительных усилий и затрат предстает как "клиент" в этих сетях), усовершенствованную мультимедиа-почту NeXTmail, возможность разделения через сеть таких устройств, как CD-ROM и факс-модемы. Наборы объектов пополнены 3D Graphics Kit, DatabaseKit, а также PhoneKit — для скоростной передачи голоса и данных через сети ISDN. Появилось важнейшее расширение объектно-ориентированной парадигмы — Distributed objects (распределенные объекты), обеспечивающие обмен сообщениями не только внутри приложения, но и между приложениями, компьютерами и сетями. На базе механизма распределенных объектов построена, в частности, NeXTlinks, мультимедиа-система гиперсвязей [11].

Подобное сухое перечисление лишь обозначает отдельные грани системы, давая слабое представление о целом. NeXT надо видеть. На нем

надо поработать. И если рассказывать о нем, то подробно и со вкусом — он заслуживает отдельного выпуска журнала.

Джобс утверждает, что система NeXT на 3-5 лет опережает Solaris, WindowsNT и Taligent. Наверное, он близок к истине. Однако, по-видимому, массовый пользователь просто не осознал, не дорос еще до этих завтрашних идей: компьютеры NeXT внедряются медленно. Во многом задавая идеологию и направление развития мира ПЭВМ, они остаются как бы образцом, эталоном, к которому следует стремиться — но не играют пока заметной роли на рынке. NeXT никак не удается выйти из заколдованного круга: малый объем продаж компьютеров — слабая заинтересованность разработчиков софтвера из-за малых тиражей — сравнительно малый выбор инструментальных и прикладных программ (хотя и высокого качества) — осторожность потенциальных покупателей — малый объем продаж. Тем не менее Джобс настроен оптимистически и намерен успешно соперничать за рынок профессиональных рабочих станций с фирмой Sun.



В последнее время, по некоторым данным, объемы продаж NeXT возрастают. Кроме того, NeXT выпускает на рынок версию NeXTstep 3.0 для 486 компьютеров. По слухам, Compaq и Dell планируют оснащать свои старшие модели этой операционной системой, а Canon уже подготовила kanji-японоязычную версию системы для своих компьютеров [12-13]. То есть у Microsoft появляется серьезный соперник на рынке ОС для процессоров Intel. (Я бы выразился еще определеннее: впервые у машин этой линии может появиться нормальная операционная система.) Этот факт, несомненно, породит множество предположений и прогнозов, поскольку впервые открывается поле прямой конкуренции между двумя легендарными личностями этого мира — Гейтсом и Джобсом. Если вспомнить к тому же, что Джобс и Гейтс фактически являются создателями, основателями мира персональных компьютеров, ситуация приобретает явный мифологический оттенок и возникает ощущение, что молот, брошенный в "Большого Брата" на презентации Macintosh в 1984 году, откован вновь, и наступает эпоха Возвращения Короля.

UNIX-станции интегрируют мультимедиа

Объявляют свои стратегии в области мультимедиа и традиционные производители рабочих станций. Операционная система UNIX, с ее развитой мультизадачностью, современными оконными интерфейсами над X-Windows: OpenLook, разработанным и поддерживаемым Sun, и OSF/Motif, который предпочитает DEC, с возможностями защиты и распределенной обработки весьма способствует быстрому и естественному внедрению мультимедиа на рабочих станциях.

Sun Microsystems, исповедующая принцип "клиент-сервер" при построении своих распределенных систем обработки информации, заявила о своем подходе к внедрению технологии мультимедиа [14]. Элементы мультимедиа, по мнению фирмы, должны значительно повысить эффективность взаимодействий внутри рабочих групп и между группами. Свои цели фирма формулирует так: предоставить платформу для интеграции технологий мультимедиа, чтобы обеспечить "People to People Communications", "для всех людей, для всех функций, в любом месте". Впрочем, эта заявленная стратегия еще не отлилась в конкретную архитектуру, вроде QuickTime или MPC, и смотрится пока скорее "декларацией о намерениях".

С другой стороны, встроенные сетевые, графические и аудиовозможности станций Sun, их высокая производительность, поддержка мультизадачности, открытость в архитектуре (SPARC) уже обеспечивают мощную и гибкую платформу для внедрения технологий мультимедиа [15]. Сочетание этих технологий с давно поддерживаемой Sun технологией ONC (Open Network Computing) фактически превращает станции Sun в "интерперсональные компьютеры" — по Джобсу.

Этому способствует и система Rapport фирмы Clarity Software, позволяющая обмениваться данными мультимедиа с компьютерами Mac и IBM PC по разнородной сети, представлять эти данные на SPARC-station в виде, привычном для пользователей этих компьютеров. Система включает набор конверторов для различных форматов данных мультимедиа, применяемых на IBM PC и Mac, а также редактор-компоновщик документов, который поддерживает интеграцию широкого спектра данных: текстов от разных текст-процессоров, электронных таблиц, оцифрованных видеоизображений, презентационной графики, сканированных документов, факсов, звуковых данных и электронной почты. Rapport помогает снять многие проблемы, связанные с использованием разнородных компьютеров в организации, обеспечить плавный переход с компьютеров одной линии на другие [16].

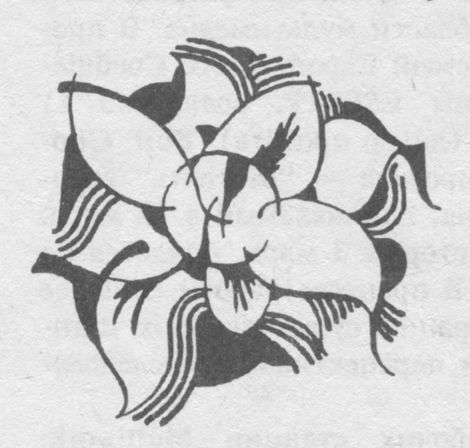
Отметим также выпущенную Sun систему VideoPix. Это набор из видеоплаты и программного обеспечения,

позволяющий захватывать и манипулировать изображениями из различных видеоисточников, вставлять видео в прикладные программы, запоминать в базах данных и т.д. Возможна работа с 24- или 8-битными изображениями, поддерживаются стандарты NTSC и PAL, JPEG-сжатие. В сети с этой платой может работать до 4 пользователей; цена системы — 900 долларов для одиночного пользователя плюс по 200 за каждого сетевого [17].

Sun выпустила еще в прошлом году целый каталог программных и аппаратных продуктов третьих фирм (более 170 наименований), обеспечивающих работу станций Sun с различными аспектами технологии мультимедиа, в разных областях деятельности, от ISDN до систем виртуальной реальности и от видеопроизводства до больничных систем [14]. Часть этих продуктов можно было видеть на COMTEK´92, где Sun, пожалуй, лучше всех предстала с точки зрения интегрированных мультимедиа-способностей компьютера (Apple в лице СП "Интермикро" систему QuickTime и ее приложения, к сожалению, не выставляла, единственный на выставке NeXT был задвинут в дальний угол экспозиции и активно не пропагандировался, а демонстрация мультимедиа на IBM-совместимых машинах в основном сводилась к 3D-графике и анимации на стендах Steepler, ComputerLand и Autodesk). В частности, запомнился демонстрационный пакет "Great Quake" фирм Parallax Graphics (создателя видеоплаты VideoView) и Paradise Software (разработчика авторских средств HyperCmedia toolkit). Это история калифорнийского землетрясения 1989 года, включающая видеофрагменты, голосовое и звуковое сопровождение, сканированные газетные отчеты и фотоматериалы. Все материалы были связаны между собой с помощью интерактивного гипермедиа-интерфейса. Пожалуй, это вообще была первая широкая демонстрация для наших пользователей мультимедиа-продукта современного уровня, возможностей, которые дает мультимедиа при построении справочных и обучающих систем.

Первые шаги на рынке мультимедиа сделала DEC. Стратегия DEC в области мультимедиа заключается в интеграции новых типов данных (аудио и видео) в общую стратегию распределенных систем, проводимую фирмой. Действуя в соответствии с этими принципами, фирма объявила в октябре 1991 года программу Distributed Multimedia и представила ряд программных и аппаратных продуктов, которые позволят разработчикам создавать распределенные, интегрированные мультимедиа-приложения. В число этих инструментальных средств входят XMedia Tools и DECmedia hardware options.

XMedia Tools ("Наборы разработчика" и "Наборы исполнения") — это программные средства, интегрирующие новые и традиционные типы данных под OSF/Motif. Составной частью этих средств является Audio-Video Run-Time Kit, предназначенный для конечных пользователей (270 долл.), содержащий драйверы, серверы, образцы программ для создания движущихся картинок, библиотеки аудио- и видео-клипов; в аналогичный набор разработчика (1000 долл.) входят дополнительно библиотеки программ, расширенные библиотеки клипов, электронная документация.



DECmedia hardware options представляет собой мультимедиа-платы расширения для RISC-станций DECstation 5000 моделей 120, 125 и 200. Сейчас доступны 24-битная графическая плата TX-Frame Buffer (5500 долл.); дочерняя плата к TX — DECvideo/PIP (Picture-In-a-Picture, картинка в картинке, 2000 долл.), которая обеспечивает демонстрацию живого видео (NTSC, PAL, SECAM) в окне X-Windows; DECaudio — звуковая плата, также 2000 долл. Платы (кроме PIP) занимают по 1 слоту TURBOchannel.

Объявлено, что целый ряд крупных фирм уже разрабатывают продукты с использованием XMedia Tools и DECmedia — это BBN, Oracle, AimTech, Imagine и др. Разрабатываемые прикладные продукты будут относиться к сферам тренинга, визуализации, телеконференций, создания деловых презентаций, видео- и аудиопроизводства, банковско-финансовых и управленческих применений. Сама DEC планирует выпустить в начале 1992 года систему распределенных видеоконференций для RISC-станций, а также FaxServer для посылки факсов с любой DECstation через электронную почту ULTRIX [18-20].

Разрабатывается также технология CMN (Community Multimedia Network) — сеть, способная передавать пакеты Ethernet со скоростью 10 Мбит/с в спектре одного телевизионного сигнала, работающая по каналам LAN и MAN (Metropolitan Area Network, высокоскоростные кабельные сети, прокладываемые в тоннелях метро). Прототип специального модема для такой сети стоил в прошлом году 50 тыс. долл., в этом — должно быть — 5 тыс. долл., а в будущем году планируется встроенная плата для компьютера ценой менее тысячи долларов. Уже проводились пробные передачи по сети кабельного телевидения на расстояние до 35 миль [6].

DEC не замыкается только на рынке рабочих станций. Например, фирма поставляет DECpc Multimedia — комплект, состоящий из PC/AT, оснащенного платой DVA-4000, и лазерного видеоплеера, управляемого этим компьютером. Это "аналогово-цифровая" система мультимедиа, позволяющая работать с интерактивными видеодисками и относящаяся к "уровню 3" по классификации систем воспроизведения видеодисков. Несомненно, Alpha, новая система компьютеров DEC, привнесет что-то новое в мир мультимедиа; у меня пока нет подробной информации о ней, но не упомянуть ее было бы неправильным.

"Конкорд" в мире мультимедиа

Ряд крупных европейских фирм решили, в качестве противовеса американским и японским разработкам, проводить свою линию в области мультимедиа. В проект Multiworks, финансируемый Европейским Сообществом в рамках программы ESPRIT, вовлечено 11 фирм, в том числе Olivetti (лидер проекта), Bull, Chorus, Acorn, Philips. Цель проекта — "отстоять" европейский рынок мультимедиа, оцениваемый в 12 млрд. долларов в 1994 году, из которых 3 млрд. придется на программное обеспечение. В процессе работы большее внимание уделяется интеграции существующих стандартов и подходов, нежели перспективным исследованиям и разработкам.

Разрабатываются 2 рабочих станции Multiwork: "авторская" станция разработчика продуктов мультимедиа MIW (Multimedia Integrated Workstation) и дешевая "воспроизводящая" станция MIW L. Они имеют разные процессоры, однако основаны на стандарте EISA и совместимы по шине с PC AT. MIW построена на 486-процессоре. Первый выпущенный компьютер MIW — это рабочая станция CP486 фирмы Olivetti. Ожидается, что другие участники проекта также выпустят собственные, взаимно совместимые, версии MIW. MIW L построен на RISC-процессоре фирмы Acorn. Операционной системой станций Multiwork будет UNIX французской фирмы Chorus Systemes, Chorus/MIX — одна из наиболее продвинутых версий UNIX, позволяющая, с рядом доработок, эффективно обрабатывать данные мультимедиа. Будет расширена файловая система UNIX, чтобы можно было считывать и записывать непрерывные массивы аудио- и видеоданных. Расширения X-Windows позволят синхронизировать эти данные, управлять ими, а также графикой и анимацией.

В состав оборудования станций входит предназначенная для совмещения видео и графики плата Movie, она располагается между графическим контроллером и монитором. Для подключения музыкальных и аудиоустройств обеспечиваются интерфейс MIDI, поддержка аудио-CD дисков; предусмотрены устройства распознавания и синтеза речи — этим должен заниматься интеллектуальный контроллер Voice/Audio Processing System. Возможно использование плоского сенсорного дисплея, позволяющего вводить рукописный текст. Доступом к дисковой памяти управляет RISC-процессор ARM3, который возьмет на себя и часть функций операционной системы. Разрабатываются специальные СБИС для работы с волоконно-оптической локальной сетью и доступа к сетям ISDN.

Объектно-ориентированная среда программирования будет базироваться на C++ и CLOS (Common Lisp Object System). Предусмотрены библиотеки объектов, представляющих физические и логические ресурсы системы. Набор инструментов Interactive Environment позволит программистам создавать в интерактивном режиме собственные графические объекты, помимо тех, которые стандартно заложены в набор объектов OSF Motif.

Процесс MMSequence позволит синхронизировать данные из различных источников, например, звуковые дорожки с видеопоследовательностями. Для этого потоки данных разбиваются на "элементарные блоки", каждый из которых содержит фиксированное количество кадров. Разрабатывается авторский и инструментальный софтвер, включающий систему Multicard (видимо, нечто подобное HyperCard), а также средства искуственного интеллекта, экспертных систем, электронных публикаций, объектно-ориентированную базу данных [21].

С.Новосельцев
Тел.: (095)938-66-10
e-mail: next@ipian15.ipian.msk.su

По материалам:

1. Ching-chin Chen. What Is Happening in the Land of Multimedia: Revealed throw Recent Meetings. Microcomputer for Information Management, June, 1991.
2. Multimedia for the Mac. ISR: Intelligent Systems Report, July, 1991.
3. M.O'Neil. Knowledge for sale. Personal Computer World, September, 1991.
4. T.McCarrol. Любовь с первого байта. ComputerWorld USSR, 8-9/1991.
5. S.Rosental. QuickTime steals the spotlight. InfoWorld, March 9, 1992.
6. C.Birkmaier. The New Digital Video. Videography, Special Edition, 1992.
7. T.Quinlan. Apple breaks Macintosh OS mold. InfoWorld, March 9, 1992.
8. S.Alsop. Apple Computer does the right thing with OCE services. InfoWorld, March 9, 1992.
9. Sculley juggles Apple's priorities with Mac, Pink. InfoWorld, March 9, 1992.
10. News. Byte, April, 1992, p.28
11. NeXTstep 3.0 Features. Материалы NeXT Computer, Inc., 1992.
12. J.R.Peck. All Aboard for PW Computing! Personal Workstation, May, 1991
13. N.Baran. NeXT to Step into the 486 Arena. Byte, April, 1992.
14. Multimedia Portfolio. Sun Microsystems, April, 1991.
15. В.Елисеев. Рабочие станции — компьютеры 90-х. Jet Microsystems, 1992.
16. Multimedia on Sun Workstations. Sun Microsystems, 1991.
17. M.Hubley. VideoPix: A Picture's Worth A Thousand Words. Personal Workstation, May, 1991.
18. Digital announces multimedia plans. ISR: Intelligent Systems Report, April, 1991.
19. B.O'Connell. Media Integration. DEC Professional, May, 1991.
20. G.Zimmerman,D.Bouffard. Digital unveils distributed multimedia tools and options for UNIX workstations. Relcom.From: minter@MR4DEC.ENET.DEC.COM Subject: [NEWS] MultiMedia Press Release 25 Oct 91 20:39:02 GMT.
21. C.Davidson. Europe United with Multimedia. Open Systems, June, 1991.

В прошлом номере мы представили нашим читателям СП "Микроинформ", ниже речь пойдет о еще одном безусловном их достижении — об учебном центре.

Век живи — век учись...

Собственно говоря, само СП началось именно как учебный методологический центр в области вычислительной техники. Тем более неожиданно, что и сегодня фирма с весьма устойчивым финансовым положением продолжает заниматься, скажем мягко, далеко не самым прибыльным делом. Впрочем, что может быть дороже благодарности одаренного знанием человека? А "Микроинформ" это делает (одаривает) весьма настойчиво и умело.

Автор этих строк сам неоднократно бывал и проходил подготовку в учебных центрах ведущих компьютерных фирм в Германии, Австрии, Венгрии (после чего эти фирмы, собственно, и стали ведущими — ред.), и его, казалось бы, трудно чем-либо удивить. Но увиденным в учебном центре "Микроинформ" он был просто потрясен: прекрасный дизайн, просторные помещения, великолепная техническая оснащенность. Например, для наглядности процесса обучения используются большие мониторы (92 см) фирмы Mitsubishi. Ученики обеспечиваются прекрасно подготовленными, лаконичными учебными материалами и методическими пособиями (в частности, при изучении ЛВС используется полный перевод фирменных пособий Novell общим объемом 600 страниц). Про преподавателей можно сказать только одно: они соответствуют окружающей технике. Девушки КомпьютерПресс, обучавшиеся на одном из курсов, вернувшись, сообщили, что им "очень понравились индивидуальные занятия, интерьер, прекрасное и чуткое отношение преподавателя". А девушкам КомпьютерПресс, поверьте, угодить не так просто!

Что касается самого учебного процесса, то он состоит из следующих курсов:

- курс для начинающих пользователей персональных компьютеров, локальных вычислительных сетей и современных средств связи;
- текстовые процессоры Лексикон и MS Word 5;
- комплексная автоматизация офисов в среде МАСТЕР;
- настольная редакционно-издательская система Xerox Ventura Publisher;
- макропрограммирование в системе SuperCalc5;
- системное сопровождение и ремонт персональных компьютеров;
- современные системы управления базами данных и языки программирования.

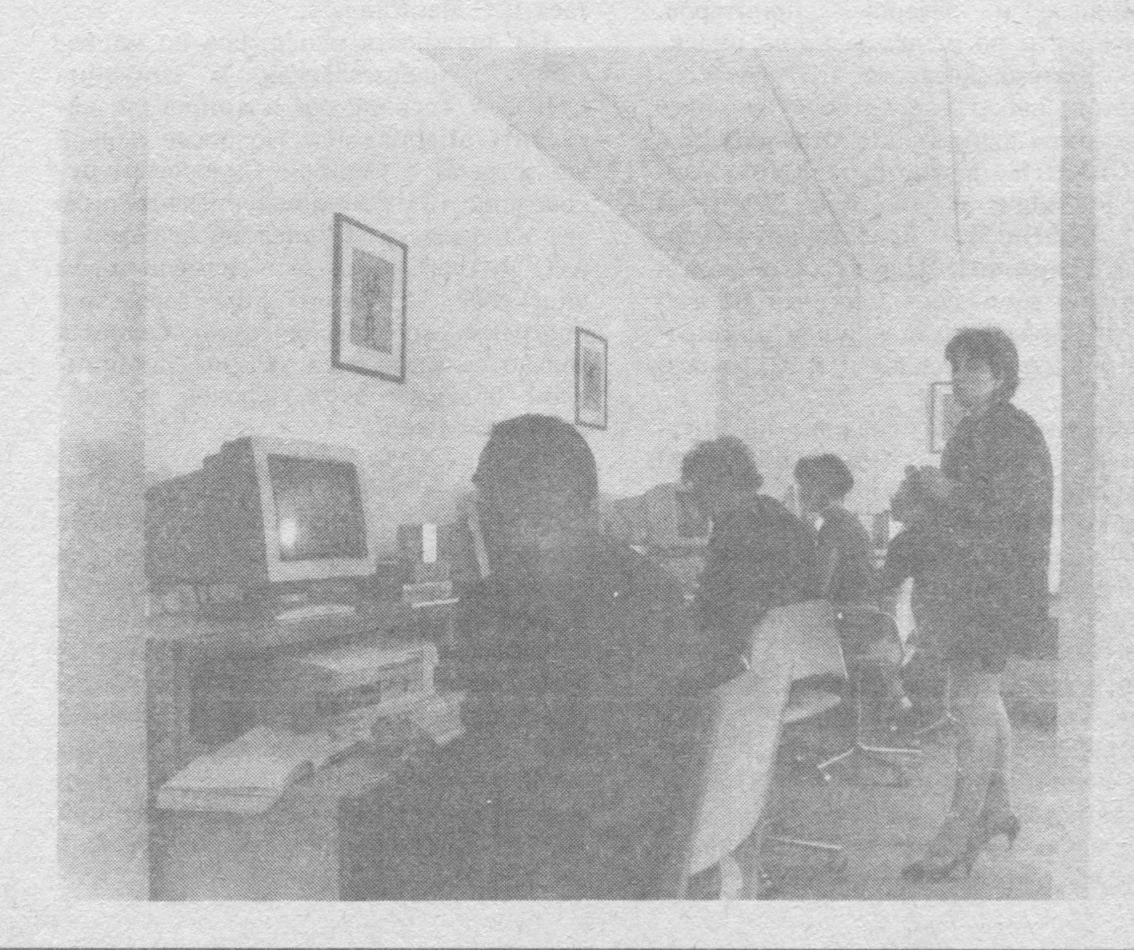
Слушатели занимаются обычно с 10 до 17 часов с перерывом на обед, который, возможно, является наиболее сильным местом курсов: там дают даже бананы!!! Но и после окончания занятий в центре алчущие знаний не будут обездолены: в собственной гостинице "Микроинформа" установлена небольшая сеть компьютеров и слушатели, проживающие в гостинице, могут вечером самостоятельно продолжить обучение.

Продолжительность обучения на курсах не превышает 15 дней. По окончании курсов выдается сертификат, который уже начинает становиться престижным. И не напрасно над дверью учебного центра висит красный стяг фирмы Novell (для завершения композиции, правда, не хватает бюста вождя, впрочем, на сертификате международного образца об окончании курсов подпись президента Novell стоит).

Стоимость полного курса обучения в СП "Микроинформ" всего лишь 10 тысяч рублей. Это разве деньги? Для сравнения — консультации самой Novell стоят в среднем 15 долларов за минуту телефонного разговора, обучение в основном европейском центре Novell в Дюссельдорфе (Германия) — около 10 тысяч долларов, сдача экзаменов — 2 тысячи долларов, при этом необходимо добавить, что затраты на каждое учебное пособие составляют в среднем 200 долларов.

У учебного центра "Микроинформа" большое будущее, только очень жаль, что таких центров в нашей стране можно пересчитать по пальцам одной руки однопалого.

И.Могучев



Новости

Регистрация нелегальных пользователей текстового редактора Лексикон

МИКРОИНФОРМ принял решение о легализации незарегистрированных пользователей ЛЕКСИКОНа. Это решение обосновано тем, что фирма признает, что в условиях правовой неопределенности, существовавшей до настоящего времени, нарушения авторских прав были неумышленными.

Всем зарегистрированным пользователям предоставляются комплекс услуг по поддержке продукта и скидки при приобретении новых версий.

В апреле 1992 года на выставке COMTEK'92 МИКРОИНФОРМ представил новую — промышленную — версию текстового редактора ЛЕКСИКОН. Она получила название ЛЕКСИКОН 1.0.

В отличие от предыдущих версий, ЛЕКСИКОН 1.0 имеет завершенный товарный вид (упаковку, документацию) и по своим функциональным возможностям во многом превосходит их. В частности, ЛЕКСИКОН 1.0 включает расширенную поддержку матричных и лазерных принтеров, возможности по использованию индексного, математического и греческого шрифтов, набор конвертеров для преобразования файлов, подготовленных в ЛЕКСИКОНе, в форматы Xerox Ventura Publisher и Microsoft Word. В ЛЕКСИКОНе 1.0 используются усовершенствованные форматы шрифтов и драйверов принтеров, так что применять эти драйверы и шрифты для предыдущих версий (6.67 и 8.92) невозможно.

Ориентируясь в основном на внутренний рынок, МИКРОИНФОРМ подготовил национальные версии ЛЕКСИКОНа 1.0 с возможностью одновременной обработки не менее 3-х различных языков в рамках одного документа. В настоящий момент МИКРОИНФОРМ поставляет армянскую, белорусскую, грузинскую, казахскую, узбекскую, украинскую версии.

К концу 1992 года планируется выпуск новой версии ЛЕКСИКОН 2.0 в рамках новой оконной технологии.

Телефон группы регистрации пользователей — (095) 233-00-06.

Здравствуйте, меня зовут Сергей! (Или что могут российские бизнесмены)

Менеджер по маркетингу СП Саммит Системс Даглас Стюарт МакКаллам возвращается в Америку. В своем выступлении перед сотрудниками московского отделения Саммит он сказал: "Я покидаю компанию в период ее взлета. Новое руководство приходит на смену не для того, чтобы спасать Саммит, но чтобы развивать успех, которого мы достигли благодаря нашим общим усилиям".

Новым генеральным менеджером СП Саммит Системс назначен Курт Кайлхакер — блестящий руководитель, с прекрасным знанием местного рынка. До настоящего времени он работал в швейцарской фирме Tranlink. Сейчас он советник правительства Москвы по вопросам приватизации, кроме того, успел поработать консультантом Центрального банка и МИД Беларуси по денежной реформе.

"...С его приходом Саммит перестанет просто зарабатывать деньги, но заставит их работать на себя...", — считает Д.С.МакКаллам.

На должность менеджера по маркетингу родительская компания CHIPS & Technologies планировала назначить американца. Но после первой же встречи с Сергеем Сидоровым руководство CHIPS единодушно одобрило его кандидатуру. Ранее он работал в А/О ДиалогСибирь — заместителем директора, а затем директором по программному обеспечению. Остается только догадываться, каким потенциалом обладает этот молодой российский менеджер.

В Москве с 28 сентября по 2 октября пройдет ежегодная конференция по компьютерной графике GraphiCon'92. В ходе конференции пройдут лекции, обучающие новым областям использования компьютеров, связанным с компьютерной графикой. Это мультимедиа, виртуальная реальность, научные системы визуализации объектов и процессов, стандарты изображений, САПР, графический пользовательский интерфейс.

Будут обсуждаться многие другие темы, например, использование компьютеров в телевидении, обучении, искусстве... Кроме того, можно будет посетить выставку компьютерных технологий и выставку компьютерного искусства.

Проводят конференцию Moscow SIGGRAPH и общество GrafO совместно с ACM SIGGRAPH и Eurographics. Телефон для справок: (095) 972-36-42.

Процесс покупки фирмы Nantucket фирмой Computer Associates, объявленный в мае, успешно завершен. Все формальности проделаны. Что дальше??

По заявлению президента Nantucket'а Лэрри Хеймендингера, покупка только ускорит разработку продукта Clipper for Windows, с кодовым названием Aspen.

Офис фирмы остался в том же месте, где и был, но несколько сотрудников все-таки было уволено.

Также не произошло пока никаких объявлений о ближних и перспективных планах CA в отношении Clipper'а. Говорят, что эти планы будут обнародованы на выставке PC Expo.

А пока CA собирается продавать Clipper под своей собственной маркой через свою дистрибьюторскую сеть и не собирается снижать цену в 795 долларов за копию.

Newsbytes News Network, June 19, 1992

Также завершились все бюрократические операции по покупке фирмы Fox фирмой Microsoft. Объявлено о планах бета-тестирования, маркетинга и поддержки продуктов Fox.

Бывший президент Fox Software Дэйв Фултон стал работать в Microsoft разработчиком концепций баз данных. На пресс-конференции он прочитал по бумажке, что, имея громадные ресурсы Microsoft, он чувствует себя, как "ребенок, у которого самая большая игрушечная железная дорога в мире".

В июле все разработчики, продавцы и обслуживающий персонал Fox переедет в Microsoft, и офис фирмы в Огайо будет закрыт.

Fox будет теперь называться очень просто: "Microsoft FoxPro" and "Microsoft FoxBASE+".

Версия FoxPro 2.5 for Windows тестируется в ограниченном числе организаций начиная с этой недели. Microsoft не забыл сказать и о том, что база данных собственной разработки Microsoft'a — Cirrus — уже находится в стадии тестирования.

Хотя новые Microsoft-Fox продукты можно заказать уже прямо сейчас, представители фирмы заявляют, что, по самым оптимистичным оценкам, они не начнут продаваться ранее конца этого года.

Newsbytes News Network,
June 29, 1992

Суд решил, что Advanced Micro Devices (AMD) не имела законного права копировать микрокод математического сопроцессора 287 фирмы Intel. Intel заявляет, что будет требовать с AMD еще и возмещения всех неполученных по этой причине прибылей.

Свое решение суд облачил в достаточно обтекаемую формулировку, от которой Intel просто пришел в победный экстаз и которую же AMD собирается в самое ближайшее время опротестовывать в суде.

Сей процесс — последний в серии судебных тяжб между Intel и AMD — начался в конце апреля этого года. Intel заявляет, что AMD украла часть разводки топологии 287 сопроцессора. Как обычно, противная сторона говорит, что ничего такого не было и в помине.

Newsbytes News Network,
June 19, 1992

Intel выпускает образцы процессора 386SL с напряжением питания 3.3 В и очень низким энергопотреблением

Фирма уверяет, что этот процессор позволит существенно увеличить время работы компьютеров-блокнотов от батарей.

По заявлению фирмы, по выполняемым функциям процессор полностью идентичен 386SX, но при этом степень его интеграции в три раза выше. Производителям машин новый процессор с тактовой частотой 20 МГц будет продаваться начиная с июля по оптовой цене 94 доллара за штуку (при минимальной партии 1000 штук), а его же версия на 16 МГц — всего за 48 долларов.

Newsbytes News Network,
June 22, 1992

Ширится круг фирм, поддерживающих новый графический интерфейс и протокол TWAIN

Этот протокол, сформулированный вначале только как предмет для дискуссии фирмами HP, Aldus, Caere, Eastman Kodak и Logitech, позволяет легко интегрировать издательские пакеты, программы обработки текста и пакеты редактирования изображений.

Протокол представляет пользователю возможность существенно упростить и унифицировать работу с самыми разнообразными устройствами ввода изображений. Просто все программы и все аппаратные средства, поддерживающие TWAIN, могут без проблем использовать информацию друг друга.

В число последователей TWAIN входят и Intel, программы распознавания символов которого Faxability Plus и Faxability Plus/OCR Software for Windows поддерживают этот протокол, и WordPerfect, заявивший, что новые версии продуктов для Windows и Macintosh будут использовать его. На сегодня протокол также поддерживают: Microtek, Calera Recognition Systems, Mustek, OCR Systems, Pentax Technologies, U-Lead Systems, Hammerlab Corporation, Orion Systems и Zedcor.

Newsbytes News Network,
June 26, 1992

Intel думает о том, а не назвать ли новый процессор, который должен быть выпущен в этом году, как-нибудь по-другому. Не 80586

Люди Intel'a заявили о полной абсурдности опубликованных слухов о том, что 586 (внутрифирменное название P5) якобы должен был быть показан на PC Expo, но что-то не заладилось. Хотя фирма еще и не выбрала имя для нового продукта, уже известно, что он будет иметь примерно 3 миллиона транзисторов на кристалле и обеспечивать производительность около 100 MIPS.

На конференции Seybold Digital World президент Intel публично поправлял всякого, кто называл новый еще не выпущенный продукт "586-м". Представительница пресс-службы Intel Нэнси Прессел сказала, что они очень обеспокоены новым веянием в придумывании названий из цифр, которые не защищаются авторским правом и законом о торговых знаках. Cyrix выпустил свой процессор "486", который с точки зрения программного обеспечения и расположения ножек полностью идентичен процессору 386 фирмы Intel. "Это вносит сумятицу в стройные ряды наших покупателей", — заявила Прессел. Она также опровергла слухи о том, что процессор загорается или расплавляется прямо на плате. Точно известно только то, что разработки таких микросхем не происходят без проблем, что выпущен P5 будет в этом году и что имя его все еще неизвестно...

Newsbytes News Network,
June 26, 1992

Фирма SHARP анонсировала компьютер с рукописным вводом, который поддерживает среду Windows. Он относится к дешевому участку рыночного спектра Японии. В продажу устройство поступит 3 июля по цене 128 000 йен (990 долларов).

The Teleputing Hotline,
May 25, 1992

Намечается борьба вокруг устройств памяти для компьютеров для работы на ходу

Исследовательская фирма Market Intelligence считает, что снижающаяся стоимость и увеличивающийся объем памяти микросхем Flash Memory ("мгновенной памяти") по стандарту PCMCIA приведет к снижению доходов от продаж жестких дисков, но к 1998 году винчестеры все еще будут преобладать с соотношением 2:3. Согласно проведенному исследованию, рынок для обеих технологий существенно вырастет — с 1 миллиарда долларов в 1991 году до более чем 12 миллиардов к 1998 году. В США микросхемы мгновенной памяти выпускают и Intel, и AT&T, и AMD, они вставляются в "дисководы" PCMCIA размером с кредитную карточку и дают сразу до 8 Мбайт памяти. Их можно менять, как дискеты, и они надежнее, чем современные жесткие диски. Но в исследовании говорится, что жесткие диски в будущем станут меньше и прочнее.

The Teleputing Hotline,
June 1, 1992

SAMSUNG представила Penmaster, компьютер-блокнот с рукописным вводом, весящий 2,1 кг. В нем используется процессор 80386SL. Когда начнутся его поставки в Великобританию, Penmaster будет оснащен MS-DOS 5.0, Pen-Windows, Penpoint и PenDOS.

The Teleputing Hotline, June 1, 1992

Компания HAYES представила новый факс-модем со скоростью 14,400 бит в секунду под названием Optima 144+ Fax 144. Розничная цена нового изделия — 519 долларов, модем отвечает стандартам V.32bis и V.42bis.

The Teleputing Hotline, June 8, 1992

Универсальный Объект Связи

Марк Эппли (Mark Eppley), председатель Traveling Software Inc., — один из подлинных новаторов в области удаленного доступа к компьютерам. Он определил термин Универсальный Объект Связи (УОС — Universal Communicating Object) как существенный элемент в "связанных компьютерных системах" ("communications-concentric computing"), новой парадигмы компьютерных систем 90-х годов, которая соединяет в себе концепцию компьютера для работы на ходу с доступом к распределенной вычислительной среде на основе ЛВС. Свой собственный проект УОС Эппли назвал Blackbird.

Blackbird уже используется в некоторых продуктах Traveling Software, например, в ведущем изделии этой фирмы LapLink Pro. Лицензию на эту технологию получила и фирма Apple — для своего компьютера Newton. По словам Эппли, УОС похож на детский кубик — законченный модуль с независимой системой связи, который можно сразу интегрировать в другое программное обеспечение. Он может определить, на каком аппаратном обеспечении он работает, и оптимизироваться в соответствии с этим, может определить, как он соединен с другим компьютером, может автоматически восстановить соединение в случае разрыва, одновременно передать файл и перенаправить вывод на принтер, он поддерживает все типы операционных систем — OS/2, UNIX, Windows и MS-DOS, а также все операционные системы с рукописным вводом. УОС может работать по кабельному соединению напрямую, через проводную коммутируемую сеть, по беспроводной сети (IBM пользуется им в своей системе PCRadio). Частично эта технология напоминает компьютерные вирусы — в том смысле, что для передачи данных с компьютера на удаленную систему LapLink Pro можно и не устанавливать. Вместо этого необходимая для работы часть кода просто будет передана с одного компьютера на другой.

Вот как происходит работа. Пользователь просит установить соединение с удаленной системой и переходит на экран Виртуального Соединения (VC), при этом вызывается специальная подпрограмма. После этого осуществляется процесс связи с операциями чтения и записи, использующими подпрограмму. Название Виртуальное Соединение не означает псевдосоединение, просто несколько соединений могут осуществляться одновременно по одному и тому же каналу передачи, при этом каждый из них считает, что имеет канал в монопольном распоряжении. Дело обстоит в точности так же, как в пакетной сети, когда каждый компьютер считает, что потоком данных пользуется только он, в то время как в действительности в сети идут вперемежку "пакеты" данных многих компьютеров.

О том, почему УОС так важен. Вы можете писать на его основе программы. Вам не надо больше беспокоиться о том, как машины соединятся. Не надо волноваться по поводу надежности связи. Просто включите УОС в стандартный набор своего арсенала для разработки программ и сконцентрируйтесь на прикладном уровне. Заявка на патент уже подана.

The Teleputing Hotline, June 15, 1992

Американский эксперт по компьютерным вирусам Ричард Левин сообщает, что с развитием технологии должны появиться **объектно-ориентированные вирусы и вирусы, использующие технологию multimedia.**

Эксперт предсказывает, что авторы вирусов скоро создадут библиотеки объектов, которые нужно будет только объединить вместе — например объект, пишущий в загрузочный сектор диска и объект, выдающий информацию на дисплей — чтобы получился полноценный вирус. И хотя все эти новые технологии работы со звуком, кино и анимацией требуют больше места, чем вирусы старых поколений, это не будет препятствием их развитию.

Newsbytes News Network, July 15, 1992

Опять крупные фирмы едят мелкие...

Тенденция к укрупнению программистских фирм продолжается. Вслед за произошедшими за последний год покупками Фокса фирмой Microsoft, Нантакета — Computer Associates, dBase — фирмой Borland объявлено о еще одной аналогичной сделке — WordStar (такой старый редактор помните?) покупает ZSoft, производителя PaintBrush'a.

Так что к Дибейсику фирмы Борланд добавится еще и ПайнтБраш фирмы ВордСтар...

Как обычно, сделка должна еще быть одобрена акционерами обеих фирм.

ВордСтар заявляет, что входит в двадцатку крупнейших продавцов программ в США. Это уже не первая попытка этой компании купить когонибудь для расширения области своей деятельности.

Как обычно, представители обеих фирм предполагают, что сделка будет обоюдно выгодной — WordStar научится хорошо работать с картинками в документах, тогда как ZSoft получит возможность расширения своих международных продаж.

К продуктам ZSoft относятся: PC PaintBrush, программа рисования и редактирования изображений; PhotoFinish, продукт для обработки и ретуширования изображений под Windows; Publisher's PaintBrush, более мощная версия старого доброго пакета; SoftType и Publisher's Type Foundry, программы разработки, генерации и редактирования шрифтов.

Newsbytes News Network, July 15, 1992

В очередном сообщении о квартальных финансовых результатах президент Intel Энди Гроув заявил, что **эра 386 машин кончается...**

Заказы на микропроцессоры "первой волны" — 386DX и SX — значительно уменьшаются, тогда как процессоры 486 и 386SL для компьютеров-блокнотов продолжают пользоваться спросом. "Windows, OS/2 и падение цен на компьютеры сильно стимулировали необходимость использования мощных процессоров",— сказал Гроув.

Newsbytes News Network, July 14, 1992

К.Чащин, С.Барабанов, О.Дергунова, И.Вязаничев

HS Listing T.233-0339

Уважаемые разработчики и пользователи программного обеспечения!

В связи с ростом затрат на обслуживание своих клиентов "HS Listing" вынужден повысить расценки за услуги.
Размещение рекламы:
- 510 руб. для индивидуальных продавцов;
- 950 руб. для организаций.

Получение информации о ПО:
- 120 руб. для частных лиц;
- 230 руб. для организаций.
Все цены даны с учетом НДС.
Правила сотрудничества с "HS Listing" опубликованы в КомпьютерПресс №№5 и 6 за 1992 год.
"HS Listing" объявляет прием заявок на публикацию материалов в раздел "HARDWARE".

АРМ

001 OK
Автоматизация работы отдела кадров предприятия.
Круг 14800р.*

002 MTS
АСУ предприятия снабжения.
Круг 107800р.

003 BLANK
Выписка бухгалтерских документов (счета, платежные поручения, платежные требования).
Круг 10450р.

004 INK
Автоматизация расчетов по "инкассо".
Круг 31900р.

005 POST
Автоматизация расчетов с поставщиками.
Круг 31900р.

006 DOCUM
АСУ "Документ" (договора и текстовые документы).
А.Шипилов 13200р.

007 АРМ бухгалтера "Mini-GROSS"
Синтетический и аналитический учет хозяйственной деятельности. Расчет баланса в любой момент, платежные документы.
PL group 16500р.*

008 АРМ брокера "КОРОНА"
Просмотр котировок в формате любой биржи. Выбор товаров по сложному запросу. Оформление договоров.
PL group 11000р.*

БАЗЫ ДАННЫХ

009 BIBL
Система управления библиографическими справочниками. Справочник по компьютерам, более 1000 ссылок.
А.Гвоздев –*

010 LAS&BASE
Программа-каталог для обработки научной и библиографической информации.
Влад И. Мир 165р.

ГРАФИКА

011 GRAPHIC INTERFACE
Экспорт, импорт и программная поддержка работы с изображениями.
HardSoft 5300р.*

ЗАЩИТА

012 PROTECTION SYSTEM
Защита .EXE-файлов от несанкционированного копирования. 3 типа защиты.
HardSoft 3400р.

013 KEYDISK
Создание ключевой дискеты и привязка к ней исполняемых программ.
HardSoft 1600р.

014 HARDLOCK
Индивидуальный и многопользовательские пароли на жесткий диск.
HardSoft 1400р.

УТИЛИТЫ

015 HS DRIVER
Система поддержки национальных алфавитов. Драйвер EGA, VGA, KBD. Редактор шрифтов, планировщик клавиатуры.
А.Аникин 2600р.

016 PRINT SCREEN
Утилита для распечатки содержимого экрана.
IPG 3300р.

ПРИКЛАДНЫЕ ПРОГРАММЫ

017 PNEUMO
Расчет потокораспределения в пневмосети высокого давления с произвольной структурой.
ТИиР –

018 MEMORY METER
Оценка объема кратковременной памяти человека. Тест для самопроверки и профессионального отбора.
В.Власов 380р.

019 TDT
Двумерные поля температур в областях с произвольной геометрией, любые граничные условия, мощная графика.
ТИиР –

ИНСТРУМЕНТЫ

020 MENU FOR PROGRAMMERS
Организация диалога типа МЕНЮ из прикладных программ (MS Fortran, MS C, Turbo C, Turbo C++, Turbo Pascal).
А.Владимиров 3800р.

ОБУЧАЮЩИЕ ПРОГРАММЫ

021 PDD
Программа обучения и проверки знаний правил дорожного движения. EGA 640x350 16 цветов.
А.Баев 7600р.*

022 KANDIDAT
Грамматика английского языка.
Златоуст 8250р.

023 TRAKTAT
Аннотирование, редактирование медицинских текстов (русский как иностранный).
Златоуст 5500р.

024 IGROK
Комплект языковых компьютерных игр.
Златоуст 6600р.

025 IGROTEKA
Комплект языковых компьютерных игр.
Златоуст 16500р.

026 VLADIMIR
Упражнения по русской грамматике.
Златоуст $220.

027 CL
Программа обучения "с нуля" популярному языку программирования Си.
М.Казенов –

Отсутствие цены ПО означает, что "HS Listing" продуктом не располагает, а '*' в конце объявления — наличие демо-версии.

Название ____________________

Текст ____________________

Автор ____________________

Прошу распространять:
☑ Информацию
☐ Демо
☐ Программу   Цена ________р.

HS Listing
Адрес: 109180, Москва, а/я 34
Р/с 1609011 в Шаболовском отд. МББ Москвы, код 2Е
МФО 201467 "Круг"

ОДИН ИЗ ВЕДУЩИХ ОТЕЧЕСТВЕННЫХ ПРОИЗВОДИТЕЛЕЙ СПЕЦИАЛИЗИРОВАННОЙ МЕБЕЛИ ДЛЯ КОМПЬЮТЕРНЫХ ЦЕНТРОВ, ОФИСОВ, БАНКОВ И БИРЖ



ДОК-17

НА ВАШЕМ РАБОЧЕМ МЕСТЕ ВСЕ БУДЕТ ПОД РУКОЙ, ЕСЛИ ВЫ ИСПОЛЬЗУЕТЕ КОМПЛЕКТЫ «КЛЕРК» И «ОФИС-МОДУЛЬ»

129128 Москва, ул. Ростокинская, 2. Телефон: (095) 191-06-43, 181-04-67
Факс: (095) 187-98-62 E-mail: dok@aquillon.msk.su

ОТЛИЧНАЯ НОВОСТЬ ДЛЯ ПРОГРАММИСТОВ!

Русская версия Clipper Tools II на российском рынке!
Ее могут приобрести только зарегистрированные пользователи *Clipper 5.0!*
Вместе с Clipper Tools II Clipper 5.0 становится еще более мощным и продуктивным!

Clipper Tools II является идеальным дополнением для тех, кто профессионально работает на Clipper.
Clipper Tools II — это расширенная инструментальная библиотека функций для Clipper 5.0.
Clipper Tools II открывает программистам Clipper новые области применения, доступные прежде лишь специалистам, имеющим большой опыт работы на языках ассемблера и Си.
Clipper Tools II включает в себя более 500 оптимизированных функций, сгруппированных по 17 тематическим направлениям, в том числе:
▪ функции поддержки окон; ▪ усовершенствованные драйверы; ▪ видеофункции; ▪ дисковые утилиты; ▪ функции поддержки последовательного интерфейса; ▪ сетевые функции; ▪ математические функции; ▪ функции непосредственного чтения и записи; ▪ функции синтаксического анализа; ▪ функции преобразования для дат и времени; ▪ системные переключатели; ▪ прочие функции.
Не успевшим приобрести Clipper 5.0 предоставляется счастливая возможность купить русифицированный комплект программного обеспечения Clipper 5.0 и Clipper Tools II за рубли!
В стоимость продукта включены техническая поддержка, журнал Nantucket News, консультации по «горячей линии» и др. услуги.
По вопросам приобретения обращаться в СП «Магнит». 127018 Москва, 2-я Ямская ул., 15.
Тел.: (095) 289-44-77; (095) 289-44-83.

